2/25/82

REPORT NO. NADC-81089-50

MIL-STD-1553 PROTOCOL SPECIFICATION

FOR P-3 MODERNIZATION

William Bermingham
Robert Wolfe
Software and Computer Directorate

FEBRUARY 1982

Phase Report
Airtask A5495490/001D/1W1150000

Approved for Public Release; Distribution Unlimited

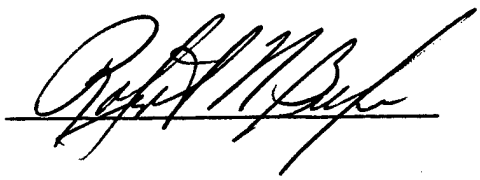
Prepared for
NAVAL AIR SYSTEMS COMMAND
AIR-240
WASHINGTON, D.C. 20360

N O T I C E S

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteenth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY:  DATE: 2/22/82

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NADC-81089-50	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: MIL-STD-1553B Protocol Specification for P-3C Modernization	5. TYPE OF REPORT & PERIOD COVERED: Phase Report
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): William Bermingham, Robert Wolfe	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Software and Computer Directorate (Code 5022), Naval Air Development Center, Warminster, PA 18974	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: AIRTASK A5495490/001D/ 1W1150000
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Air Development Center, Command Projects Directorate (Code 1P61), Warminster, PA 18974	12. REPORT DATE: February 1982
	13. NUMBER OF PAGES: 78
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Naval Air Systems Command (PMA-240), Department of the Navy, Washington, DC 20361	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approval for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

MIL-STD-1553B	Data Bus
Bus Protocol	Communication Protocols
Digital Multiplex Bus	Layered Protocol

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes a protocol developed for the P-3C Modernization Program utilizing the MIL-STD-1553B Data Bus. The protocol described in this report is intended as a menu of tools to be utilized, as the interfacing requirements govern, by the I/O and applications programmers. The word formats specified are a layer below that of the application software except where specifically mentioned. This protocol is intended to satisfy all currently envisioned P-3C interfacing requirements which fall into the realm of the restrictions and capabilities imposed by MIL-STD-1553B.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

1.0 Scope

This specification establishes the performance requirements of the P-3C Low Data Rate Bus Protocol as applicable to MIL-STD-1553B.

2.0 Applicable Documents

2.1 Specifications

Military

MIL-STD-1553B Aircraft Internal Time Division Command/Response Multiplex Data Bus

P-3C Low Data Rate Bus Protocol Requirements

3.0 Requirements

3.1 General

The requirements imposed upon this P-3C 1553B Bus Protocol are multi-level and complex. The short term requirements are to satisfy the interfacing needs of the P-3C Modernization Communication Subsystem development. The longer range requirements address supporting a peripheral bus structure for all P-3C peripheral devices. Some of the interfacing characteristics which demonstrate a wide range of needs are buffer size, response time, interrupt capability, local intelligence and data priorities. It is the intention of this document to provide a layered architectural approach to the protocol such that each device will implement just enough of the protocol to satisfy its particular interfacing requirements. This will insure that both hardware and software design are only as complex as the particular peripherals' need. For details of the requirements refer to NADC document "P-3C Low Data Rate Bus Protocol Requirements" dated 16 September, 1980.

3.2 Constraints

3.2.1 1553B - All transfers on the P-3C 1553B data bus shall comply fully with MIL-STD-1553B and the protocol set forth in this specification.

3.2.2 System - The system constraints are as described in the "P-3C Low Data Rate Bus Protocol Requirements" document dated 16 September, 1980. At the time of writing of this document, little or nothing was known of the specific requirements imposed by the P-3C Modernization Communication Subsystem development. It is intended that the above referenced requirements document encompasses all P-3C applications.

3.3 Protocol Performance

3.3.1 General

The 1553B Multiplex Data Bus specifies an electrical interface and some very low level requirements for accomplishing data transfers; therefore, each user must determine a bus protocol which addresses many unspecified areas. The object of the bus protocol definition specified herein is to enhance the 1553B control function by providing a means to exchange timely interrupt information, assign priority levels, present positive acknowledgment of receipt of data, develop modularized and transportable software modules (both Remote Terminal and Bus Controller), add or delete terminals without affecting all other terminals' software, and increased interface integrity and reliability.

3.3.2 Capability

The bus capability which is available for application I/O transfers is the 1553B bandwidth less the protocol overhead. In order to optimize this performance, a flexible protocol structure is required where each unit initiates its bus scheduling while also allowing a priori data exchange requirements within the bus controller. The benefits of this are that it (1) allows full flexibility in accommodating various system configurations and loads, (2) facilitates an adaptive bus scheduling scheme which allows most efficient bus utilization with asynchronous data, and (3) promotes bus controller and remote terminal software modularization.

3.3.3 Flexibility

The protocol shall easily accommodate additions, deletions, and substitutions of hardware units or application software modules that are within the channel capacity, with minimal impact. As the system changes, the associated software modifications shall be restricted to the units directly involved. Such modifications shall be transparent to the remaining units of the system. Therefore, the protocol shall have a performance capability to accommodate the worst case data rates and response times of foreseeable 1553B candidate subsystems.

3.3.4 Availability

In order to allow degraded mode operations, the protocol scheme shall incorporate features such that its software/hardware is not subject to a single point failure. All error recovery features of the 1553B bus shall be implemented in the protocol with operator override available for bus selection and bus controller allocation.

3.3.5 Characteristics

3.3.5.1 Bus Controller

Control of the bus shall be centralized within the bus controller. Any terminal which has sufficient reserve processing capability and memory is a candidate for bus controller. Bus controller software may be replicated or available from mass storage for those units having bus controller capability. The bus control software shall be functionally identical regardless of which unit is bus controller. This structure simplifies bus controller design, especially in the case of error management, since the control is centralized rather than distributed.

3.3.5.2 Polling

The bus controller shall assess bus demand of all the bus users on a periodic basis by polling. The polling rate for each user shall be software definable and modifiable via the bus controller software. The poll responses are tabulated to form part of the bus data base.

3.3.5.2.1 Poll Requests

With a bus as an I/O channel and using a generalized approach, it is necessary to prefix data exchanges with information to describe which bus user originated the transfer. In 1553B, only the bus controller has source and destination information concerning each I/O transfer, therefore the protocol shall provide this information to each bus user prior to executing a transfer.

3.3.5.2.2 Scheduling

The bus controller shall operate on the poll response data base and incorporate new inputs into its current BC Control Table (section 3.6) according to the priority of the poll responses. The bus controller shall insert its application I/O request directly into the BC Control Table since it does not poll itself.

3.3.5.3 Remote Terminal

A remote terminal is any bus unit which is not operating as the bus controller or as a bus monitor (as per MIL-STD-1553B). There are two classes of RT's defined by this protocol and therefore capable of operating on this bus.

3.3.5.3.1 Polled RT's

This is a class of RT whose bus demands shall be determined by the bus controller on a periodic basis by polling. The RT shall respond to a poll from the bus controller with the Request CDW described further on in this document (section 3.5.5).

3.3.5.3.1.1 Minor Cycle

This is the minimum period in which an RT can be polled. For this protocol it is 10 ms.

3.3.5.3.1.2 RT Cycle

This is the cyclical period in which an individual RT is polled. For this protocol each RT should be polled at a rate which is an integer multiple of the minor cycle.

3.3.5.3.1.3 Major Cycle

This is the time required to poll all RT's at least once. It shall be equal to the maximum RT cycle implemented in the system.

3.3.5.3.2 Unpolled RT's

This is a class of RT whose only bus transfers are upon request by a polled RT, the BC, or are predefined, cyclic transfers with a known RT or the BC. This permits the RT to execute an Information Transfer Sequence (section 3.8.3) without being polled. In this way the complexity of the RT need only to implement those sections of the protocol essential to its operation. This class of RT shall only be required to implement the Information Transfer Sequence and the applicable Test Sequences. The intention is that the RT's hardware implementation would not require a host processor.

This class of RT's shall not have the capability to perform nesting or initiate chaining (see paragraphs 3.4.1.4 and 3.4.1.6). If this RT is part of a chained transaction, it shall never be polled during the chain. It is intended that this class of peripherals shall always be ready to transmit or receive information, as required. If a particular peripheral of this class requires it, the use of the BUSY bit is not precluded.

For transmit mode only this class of RT's may implement the retry function in one of two ways. The preferred method is to maintain the old output buffer until a new output command is received (double buffering). The alternative method is that a retry generates the newest data available, not necessarily the same data buffer.

3.3.5.4 Protocol Status

Since the protocol is the vehicle by which control is exercised over the bus, it follows that the protocol shall provide the means of maintaining the bus system status. This status shall include one level of current, pending, suspended or complete I/O status of each unit. Error recovery, hardware/software development, and integration problems are minimized by having bus characteristics readily available with a comprehensive status data base.

Specific required characteristics of the protocol status include:

a. Positive feedback to ensure proper I/O transfers
b. Test software for checking bus integrity
c. Real-time maintenance of transfer requests and subsequent states for all RTs

3.3.5.5 Information Transfer

Information transfer between all terminals (the BC and all RTs) shall take place on a coequal basis. The polling and scheduling results shall use the same prioritization scheme for determining when the RTs and the BC gain bus access for information transfer.

3.3.5.6 Loop Test

The loop test shall be executed to (1) verify channel integrity upon initialization, (2) determine whether channel malfunctions indicated by error interrupts are transient or hard failures.

Testing shall be an integral protocol function rather than a subset of application functions.

3.3.5.7 Error Handling

Error handling shall be implemented through an error retry technique. Upon detection of an error, protocol or 1553B, the BC shall retry the sequence a maximum of two times. If a sequence is unsuccessful on two consecutive retry attempts (three consecutive errors including the original) then the BC shall utilize the Bus Reconfiguration function (section 3.3.5.8).

3.3.5.8 Bus Reconfiguration

This shall include the strategy for implementing 1553B recovery techniques which include (1) bus switching and (2) reassignment of the bus controller function.

This strategy shall provide operator status when invoked and be subject to operator override. (section 3.9.4)

3.4 Protocol Description

3.4.1 Protocol Definitions

3.4.1.1 Message - (as per MIL-STD-1553B)

A message is the transmission of a command word by the bus controller, transmission of a status word by the addressed RT, and, if they are specified, transmission of from 1 to 32 data words in the appropriate direction. For RT-to-RT, this shall include two commands, one to each RT, two status words, one from each RT, and from 1 to 32 data words from one RT to the other.

3.4.1.2 Block

A block shall be the transmission of a message that includes a data word or data words.

3.4.1.3 Transaction

A transaction is the complete exchange of information (protocol and/or data) between bus units whether BC-to-RT, RT-to-BC, or RT-to-RT. A transaction may consist of one message or many messages. See Section 3.9.2 for transaction examples.

3.4.1.4 Chaining

Chaining is a method in which the RT being polled tells the BC one of three things to do when the current transaction is complete: 1) poll requestor next, 2) poll acceptor next, 3) terminate chain. The requestor is the RT being polled and the acceptor is the RT with whom the requestor needs an exchange of information. Chaining is meant to be used between two RT's which have predefined series of transfers to perform between them, in any combination of directions, which can be accomplished with minimal protocol overhead and in the least amount of time.

3.4.1.5 Linkage

When there is a transfer of information between two terminals on the bus (i.e., a transaction is underway) then a linkage is said to exist between the two terminals (links). Linkages exist for single or multiblock transactions. For a chain of transactions the linkage exists from the first transaction until the termination of the chain.

A normal linkage exists for an RT when that RT has a transaction active with another RT. Each RT shall have only one normal linkage active at a

time. Each RT shall also be allowed to have one nested linkage (see section 3.4.1.6) active at any time that a normal linkage is active. All linkages shall be maintained while all RT's of the system are polled at their normal rate.

3.4.1.6 Nesting

Nesting is the mechanism by which a high priority unchained transaction to or from a linked RT can be interleaved with an ongoing transaction or chain of transactions. The other RT can be linked or unlinked.

3.4.1.7 Requestor

A requestor shall be defined by this protocol to be a terminal which has a bus request indication active to the bus controller at the time it is being polled. The requestor can be the transmitter or receiver of data.

3.4.1.8 Acceptor

An acceptor shall be defined by this protocol to be the terminal with which the requestor requests an exchange of information. The acceptor can be the transmitter or receiver of data.

3.4.2 Word Definitions

3.4.2.1 Command Word (CW)

as per 1553B.

3.4.2.2 Status Word (SW)

as per 1553B.

3.4.2.3 Data Word (DW)

The data word is as per 1553B but its utilization is further defined to include protocol control functions.

3.4.2.3.1 Control Data Word (CDW)

CDW's are dedicated to the overhead function of the protocol and may consist of one or three contiguous DW's depending on the function. They include (1) Request CDW's which are three DW's that respond to bus controller polling, (2) Suggest CDW's which are three DW's that preface application data with transfer information to the receiving RT while allowing the RT to prepare for the transfer, and (3) Protocol Comand CDW's which are single DW's which implement discrete type functions of the protocol in the form of a Command.

3.4.2.3.2 Test Data Word (TDW)

TDW's are used by the bus controller to test bus integrity. They comprise worst case test patterns for Manchester transmissions. The bus controller verifies data paths by transmitting TDW's from a source to a destination of a data path and recalling and comparing the TDW's received at the destination with the original TDW's. Such transfers are transparent to application level software since the RT units do not process but merely provide storage for TDW's.

3.4.2.3.3 Interrupt Data Word (IDW)

IDW's are dedicated for application program usage. It consists of two DW's (32 bits). The format and information within an IDW is entirely at the discretion of the application program; however, it is intended for important or prefacing information that requires immediate attention; e.g., application commands, status, requests or attention and definition of the content of a following block of NDW's. Only one IDW may be transferred within a block.

3.4.2.3.4 Normal Data Word (NDW)

NDW's are dedicated for application program usage. It is equivalent to one DW (16 bits). The format and information within an NDW is entirely at the discretion of the application program; however, it is intended for bulk information transfers.

3.5 Word Formats

3.5.1 General

This section defines the word formats required to implement the P-3C Modernization 1553B protocol. These word types are: The CW and SW as presently defined in MIL-STD-1553B, the CDW and the TDW which provide additional control and testing for the interprocessor protocol, and the IDW and NDW which contain the information to be exchanged by the application programs of the processors. Figure 3.5.1-1 shows the relationship between the MIL-STD-1553B word format and the protocol word format.

3.5.2 P-3C Modernization 1553B Command Word Implementation

3.5.2.1 General

The implementation of the Command Word is exactly as defined in MIL-STD-1553B. Figure 3.5.2.1-1 shows the CW format and Section 3.5.2.2 through 3.5.2.5 define the usage of each of the data fields.

Word		Bits																			
Protocol Words		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
		(CW, SW, or DW)																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
MIL-STD-1553B	SYNC			(CW, SW, or DW)																P	

Figure 3.5.1-1
Word Format for Protocol and MIL-STD-1553B

BIT	15 14 13 12 11	10 9 8 7 6 5	4 3 2 1 0
MEANING	REMOTE TERMINAL ADDRESS	T/R SUB-ADDRESS/MODE	DW COUNT/ MODE CODE

FIGURE 3.5.2.1-1

COMMAND WORD FORMAT

3.5.2.2 Command Word Bit Definitions

3.5.2.2.1 Remote Terminal Address (Bit Times 4 through 8)

As per MIL-STD-1553B.

3.5.2.2.2 Transmit/Receive Bit (Bit Time 9)

The use of this bit is exactly as defined in MIL-STD-1553B.

3.5.2.2.3 Sub-address/Mode

The Sub-address/Mode code uniquely defines the format of the data words of that particular block. The sub-address definition for mode commands (sub-address 0 and 31) is retained from MIL-STD-1553B. Sub-addresses 14 thru 19 shall be used by the BC to read the RT's Control Table. The use of sub-address 16 shall be used for polling an RT's normal requests and sub-address 17 for the nested request. A command using sub-address 18 shall be transmitted by the BC along with a data word (Protocol Command CDW, Section 3.5.7) to carry a discrete Protocol function. Also, all of these sub-addresses, 14 thru 19, can be used to read as much of an RT's Control Table, depending on the sub-address and word count, as necessary for test or diagnostic purposes (see Section 3.7).

Sub-addresses 4, 5, 6, 20, 21, 22, 24, 25, and 26 shall be used by the BC in the command word to inform an RT of the information block to be transmiited or received by that RT. In the case of an error during a block the BC shall use one of the retry sub-addresses, 7, 23, or 27, to repeat the last block.

If an RT detects a sub-address which the RT does not implement or which is illegal in this protocol, then that RT shall set the Message Error bit in its Status Word (see Figure 3.5.2.2.3-1).

3.5.2.2.4 Word Count/Mode Code (Bit Times 15 thru 19)

The Count/Mode field is used exactly as defined in MIL-STD-1553B, except that the Dynamic Bus Control mode code shall not be used. The other mode codes shall be utilized dependent upon the individual RT implementation.

Sub-address	Meaning	Comments
00000 (0)	Mode Command	
00100 (4) 00101 (5) 00110 (6) 00111 (7)	Transfer Suggest CDW, 0-29 NDW Transfer Suggest CDW, 2 IDWS, 0-27 NDWs Transfer 1-32, NDWs Retry last block (4, 5, or 6)	for nested, unchained, single or multi-block transfer to be executed
01110 (14) 01111 (15)	Transmit Normal Suggest CDW Transmit Nested Suggest CDW	BC can read RT's Control Table
10000 (16)	Transmit Normal Request CDW	read of RT's Control Table at poll time
10001 (17)	Transmit Nested Request CDW	read of RT's Control Table for nested requests
10010 (18)	Receive Protocol Command CDW	discrete protocol command to RT
10011 (19)	Transmit Last Command (1553B)	BC can read RT's last received command
10100 (20) 10101 (21) 10110 (22) 10111 (23)	Transfer Suggest CDW, 0-29 NDWs Transfer Suggest CDW, 2 IDWs, 0-27 NDWs Transfer 1-32 NDWs Retry last (20, 21, or 22)	for normal, unchained, single or multi-block transfer to be executed
11000 (24) 11001 (25) 11010 (26) 11011 (27)	Transfer Suggest CDW, 0-29 NDWs Transfer Suggest CDW, 2 IDWs, 0-27 NDWs Transfer 1-32 NDWs Retry last block (24, 25, or 26)	for normal, chained, single or multi-block transfers to be executed
11110 (30)	Transfer Test Data Words	TDWs to be used during Loop Test Sequence
11111 (31)	Mode Command	
All others	Illegal	

Figure 3.5.2.2.3-1
CW Sub-address Utilization

3.5.3 P-3C Modernization 1553B Status Word Implementation

3.5.3.1 General

The implementation of the Status Word is exactly as defined in MIL-STD-1553B. Figure 3.5.3.1-1 shows the status word format. The following section defines the usage of each of the fields.

3.5.3.2 Status Word Bit Definitions

3.5.3.2.1 Terminal Address

As per MIL-STD-1553B, also see CW implementation.

3.5.3.2.2 Message Error

As per MIL-STD-1553B.

3.5.3.2.3 Instrumentation

Always 0.

3.5.3.2.4 Service Request

Indicates to the BC that a Nested Request condition exists. This bit shall be set when the Request condition is set active by the RT and reset by the Reset Mode Command or the execution of the requested transfer.

3.5.3.2.5 Reserved

Always 0.

3.5.3.2.6 Broadcast Command Received

As per MIL-STD-1553B, also see Broadcast Sequence implementation, section 3.8.5.

3.5.3.2.7 Busy

As per MIL-STD-1553B; indicates the RT is temporarily unable to perform the command.

3.5.3.2.8 Subsystem Flag

Indicates a protocol error (soft) that is in violation of the protocol defined by this specification. Errors occurring or detected by the RT shall cause the Subsystem Flag bit to be set within the Status Word.

3.5.3.2.9 Dynamic Bus Control Acceptance

Always 0.

3.5.3.2.10 Terminal Flag

Indicates a firmware or RT error that is in violation of MIL-STD-1553B. Errors occurring within or detected by the RT interface unit shall cause the Terminal Flag bit to be set within the Status Word.

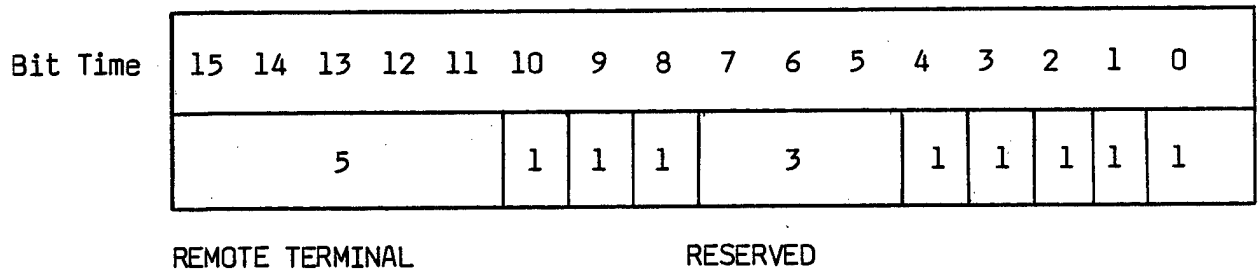


FIGURE 3.5.3.1-1

STATUS WORD DEFINITION

3.5.4 P-3C Modernization Data Word Implementation

3.5.4.1 General

The MIL-STD-1553B data word has been further defined to handle the P-3C Modernization protocol implementation. The data word usage has been categorized into several types whose format and usage are described in the following sections:

a) Request CDW (consisting of 3 data words)
b) Suggest CDW (consisting of 3 data words)
c) Protocol Command CDW (consisting of 1 data word)
d) Interrupt Data Word (consisting of 2 data words)
e) Normal Data Word (consisting of 1 data word)
f) Test Data Word (consisting of 1 data word)

3.5.5 Request CDW

3.5.5.1 General

The Request CDW shall be used to request an information transfer. It is transmitted to the BC during a poll sequence via a command with a sub-address of 16. If an RT implements nesting, a separate Request CDW shall be accessible via a command with a sub-address of 17. Both the normal and nested Request CDW's shall be contained in the RT's Control Table. The format and field definitions of the Request CDW are defined below. It consists of exactly three DW's.

3.5.5.2 Request CDW Format

Bit Time	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st DW Fields	A			B				C		D		E			F	
2nd DW Fields		G									H					
3rd DW Fields					J					K				L		

Field A: Transfer Code

VALUE	MEANING
0	NO REQUEST
1	REQUEST
2	REQUEST ACKNOWLEDGED
3	REQUEST COMPLETE/TERMINATED

This field indicates the current status of an RT's request. If there is no request this field shall be set to 0 by the RT. If the RT has a transfer request pending it shall set this field to 1 in order to be read by the BC during the poll sequence. The RT shall set this field to 2 when it receives the appropriate transmit or receive command from the BC.

If the requestor is the transmitter of the information then the BC shall update Fields A and B of the Request CDW in the RT's Control Table to a Transfer Code of 3 and the proper Completion Status, respectively. Otherwise, if the requestor is the receiver then it shall update these fields after transmitting a normal Status Word in response to receiving the information. After this the RT shall set Field B to 0 and Field A to 0, or 1 if a new request is present.

Field B: Completion Status

This field indicates the condition in which the request was completed. It is updated by the BC at the same time the BC sets Field A to 3. The RT shall then interpret this field to determine the status of the last transaction. The following table defines the values and meanings for this field.

VALUE	MEANING
0	Normal Completion (No errors in Status Words)
1	RESERVED
2	RESERVED
3	RESERVED
4	RESERVED
5	RESERVED
6	RESERVED
7	Transaction Terminated Via Protocol Request
8	Request Canceled - Addressee Linked (Normal)
9	Request Canceled - Addressee Linked (Nested)
A	Request Canceled - Addressee Busy
B	Request Canceled - By Addressee
C	Request Canceled - Insufficient Priority
D	Request Canceled - Protocol Error
E	Request Canceled - Hardware Error
F	Request Canceled - Addressee not in system

Field C: Protocol Request

An RT shall use this field to cancel all requests (normal and nested) from this RT, terminate any active linkages (normal or nested) with this RT, or terminate only a normal linkage active on this RT.

The BC shall always use the TSS (section 3.8.2.3) to acknowledge this request. If processed without any errors, then Field A shall have a 3 written into it and Field B shall have a 7. Otherwise, Field A shall have a 3 written into it and Field B the appropriate error code.

VALUE	MEANING
0	No Action
1	Cancel all requests from this RT
2	Cancel all linkages to this RT
3	Cancel normal linkage to this RT

Field D: Sequence Chaining

This field indicates whether the request is part of a chained sequence or an unchained sequence. If it is chained, the requestor informs the BC which RT in the chain shall be polled next. This eliminates a poll sequence to one of the RT's involved in the chained transfer. The following table lists the values and meanings for this field.

VALUE	MEANING
0	Unchained Transaction
1	Chained - Poll Requestor Next
2	Chained - Poll Acceptor Next
3	End of Chain

Field E: Transmit/Receive

"0" - Requestor (polled RT) to receive data from the address specified in Field G

For this case, the transmitting RT receives only the transmit Command Word from the BC. This RT does not know which RT is the receiver (the only information given to the transmitter is the sub-address and word count). This situation is similar to the unpolled RT class which transmits upon command. For specifics, refer to transaction #4 of Figure 3.9.2.5-2.

"1" - Requestor (polled RT) to send data to address specified in Field G

Field F: Sequence Code

This field shall contain the code corresponding to the first block of information to be transferred within the Information Transfer Sequence. This field shall be encoded with one of the following values: 20, 21, or 22, for an unchained, normal sequence; 24, 25, or 26 for a chained, normal sequence; or 4, 5, or 6 for an unchained, single block, nested sequence (see Figure 3.5.5.2-1). The retry codes shall not be used by the RT. These shall only be used by the BC after an error has been detected in one of the RT Status Words.

Field G: RT Address

Address of RT with which the requestor wants to exchange information.

Field H: Normal Data Word Count

This field is a binary representation of the number of NDW's to be transferred within this transaction. The count can range from 1 to 2048 NDW's. The count does not include CDW's or IDW's. All 1's equals a count of 2047, all 0's equals 2048. This field shall remain static during the transaction.

Sequence Code	Meaning	Comments
00100 (4) 00101 (5) 00110 (6)	Suggest CDW, 0-29 NDWs Suggest CDW, 2 IDWs, 0-27 NDWs 1-32 NDWs	indicates block to be transmitted for nested, unchained, single block transaction
10100 (20) 10101 (21) 10110 (22)	Suggest CDW, 0-29 NDWs Suggest CDW, 2 IDWs, 0-27 NDWs 1-32 NDWs	indicates first block to be transmitted for normal, unchained, single or multi-block transaction
11000 (24) 11001 (25) 11010 (26)	Suggest CDW, 0-29 NDWs Suggest CDW, 2 IDWs, 0-27 NDWs 1-32 NDWs	indicates first block to be transmitted for normal, chained, single or multi-block transaction
All others	Illegal	

Figure 3.5.5.2-1

Sequence Code Table

(This table is a subset of the Sub-address Table Figure 3.5.2.2.3-1)

Field J: Data Type

An 8 bit code specifying the type of data to be transferred within the sequence. The bus controller shall assign a priority level to each data type. If the data type in this field is not associated with the RT of Field G then the BC can override the address in Field G and execute a transaction with the RT associated with that data type. This function is system dependent.

Field K: Chaining Sequence Identifier

This 3 bit field defines which particular predefined chain is currently active between the two devices. Each device shall have a priori knowledge of the chain to use this field. (Additional chain information can also be contained in an IDW or NDW).

Field L: Last Transaction Status

This field shall be used to present amplifying information on error conditions occurring in the last transaction. The BC can extract this information by utilizing the Poll Request Sequence (PRS).

VALUE	MEANING
0	Normal
1	Normal Data Word (NDW) Count Error
2	Protocol Error
3	Illegal Subaddress for this RT
4	1553B Validity Error (ILL Parity, ILL Sync, Word Length Error, Word Count Error, Invalid Manchester, etc.)
5-63	Reserved

3.5.6 Suggest CDW

3.5.6.1 General

The Suggest CDW shall be used by a terminal to preface an information transfer to another terminal. It consists of exactly three DW's. Four types of Suggest CDW's shall be contained in the RT Control Table (See Section 3.7). These are the transmitted and received Suggest CDW's and the transmitted and received Suggest CDW's for a nested transaction, if the RT implements nesting. An RT shall only have a transmitted Suggest CDW or a received Suggest CDW active at a time, but not both. This is because in a transaction the Suggest CDW can only be in one direction. The format and field definitions are defined below.

3.5.6.2 Suggest CDW Format

Bit Time	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st DW Fields	A		B				C		D		E	F				
2nd DW Fields	G						H									
3rd DW Fields	J								K			L				

Field A: Transfer Code

VALUE	MEANING
0	No Request (Not used in Suggest CDW)
1	Request (Not used in Suggest CDW)
2	Request Acknowledged
3	Request Complete

When the Transfer Suggest CDW is transmitted this field shall always be set to 2. If the RT is the acceptor in the transaction then the BC shall update Fields A and B in the Suggest CDW (subaddress 14 for a normal suggest and subaddress 15 for a nested suggest) in the RT's Control Table to a Transfer Code of 3 and the appropriate Completion Status via the Transfer State Sequence. Otherwise, the receiving RT is responsible for updating these fields of its received Suggest CDW in the RT Control Tables upon completion of the transaction.

Field B: Completion Status

This field indicates the condition in which the suggest was completed. This field is updated to the appropriate status by each RT involved in the transfer at the same time that Field A is set to 3.

VALUE	MEANING
0	Normal Completion
1	RESERVED
2	RESERVED
3	RESERVED
4	RESERVED
5	RESERVED
6	RESERVED
7	RESERVED
8	Suggest Canceled - Addressee Linked (Normally)
9	Suggest Canceled - Addresse Linked (Nested)
A	Suggest Canceled - Addressee Busy
B	Suggest Canceled - By Addressee
C	Suggest Canceled - Insufficient Priority
D	Suggest Canceled - Protocol Error
E	Suggest Canceled - Hardware Error
F	Suggest Canceled - Addressee not in system

Field C: Protocol Request

This field is not used for a Suggest CDW.

Field D: Sequence Chaining

This field indicates whether the suggest is part of a chained sequence or an unchained sequence. It shall be encoded the same as the Field D of the

Request CDW.

VALUE	MEANING
0	Unchained Transaction
1	Chained - Poll Requestor Next
2	Chained - Poll Acceptor Next
3	End of Chain

Field E: Transmit/Receive

This field shall always be set to "0" indicating that the other terminal is to receive data.

Field F: Sequence Code

This field shall contain the code corresponding to the first block of information being transferred within the Information Transfer Sequence. This field shall be encoded with one of the following values: 20, 21 or 22 for an unchained, normal sequence; 24, 25, or 26 for a chained, normal sequence; or 4, 5, or 6 for an unchained, single block, nested sequence (see Figure 3.5.5.2-1).

The retry codes shall not be used by the RT. These shall only be used by the BC after an error has been detected in one of the RT Status Words.

Field G: RT Address

This is the address identifying the terminal transmitting this Suggest CDW.

Field H: Normal Data Word Count

This field is a binary representation of the number of NDW's to be transferred within this transaction. The count can range from 1 to 2048 NDW's. The count does not include CDW's or IDW's. All 1's equals a count of 2047, all 0's equals 2048.

Field J: Data Type

An 8 bit code specifying the type of data to be transferred within the sequence. The bus controller shall assign a priority level to each data type.

Field K: Chaining Sequence Indentifier

This 3-bit code identifier which particular predefined chain is being initiated between these two RT's. Each device shall have a priori knowledge of the chain to use this field. (Additional chaining information can also be contained in an IDW or NDW).

Field L: Last Transaction Status

This field is not used by the Suggest CDW.

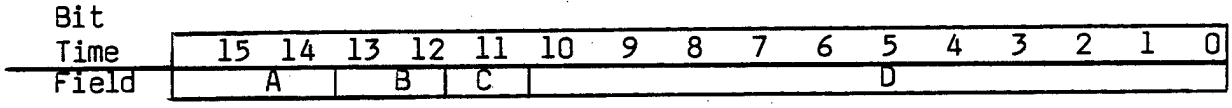
3.5.7 Protocol Command CDW

3.5.7.1 General

The Protocol Command CDW provides the BC with additional protocol control capability. The RT may or may not use the Transfer State sequence (section 3.8.2.3) to acknowledge the command. The use of the Transfer State sequence to acknowledge this command shall be determined by the device utilization. Upon receiving this word, the RT shall store it in the RT's Control Table to be updated accordingly. Field A of this CDW shall be updated and transmitted back to the BC upon request.

The Protocol Command CDW is a single sixteen bit DW and is transmitted to the RT using sub-address 18. The RT shall be responsible for updating the Response field, if required, and shall transmit this CDW to the BC upon request, also via sub-address 18.

3.5.7.2 Protocol Command CDW Format



Field A: Response Field

This field shall be set to zero when transmitted to the RT. The RT shall store the word in the RT's Control Table and maintain progression of this field to 1 or 2 as an acknowledgement to a Protocol Command CDW.

VALUE	MEANING
0	No Command
1	Command Acknowledged
2	Command Completed - This value shall be set when the command has actually been completed. Until then the value 1 shall be returned when requested by the BC.
3	Illegal

Field B: Protocol Command

VALUE	MEANING
0	No Action
1	Cancel all Requests
2	Cancel all Linkages
3	Cancel Normal Linkages

Field C: Reset Service Requests

"0" - No action
"1" - Reset Service Request Bit

Field D: Undefined - set to all zeroes

3.5.8 Interrupt Data Word (IDW)

3.5.8.1 General

The format and utilization of the IDW shall be at the discretion of the application program (see Section 3.4.2.3.3). The IDW shall be interpreted in conjunction with a sub-address or Sequence Code of 5, 21, or 25.

3.5.9 Normal Date Word (NDW)

3.5.9.1 General

The format and utilization of the NDW shall be at the discretion of the application program (see Section 3.4.2.3.4). The sub-address and Sequence Code fields shall indicate which data words to interpret as NDW's.

3.5.10 Test Data Word (TDW)

3.5.10.1 General

The format and utilization of the TDW shall be as defined in Section 3.8.4.2.

3.6 BC Control Tables

Complete status of all bus operations and of each RT shall be continually monitored and updated for the bus controller to maintain and control operations on the bus efficiently. To accomplish this the bus controller shall contain a data base with sufficient information of all requests, transactions, and each RT's status.

The Normal Request Table, Figure 3.6-1, contains the status of all requests in the system. After being polled, if an RT has a request it is logged into the table. Each request shall then be processed in order of priority. When a requested transaction, or chain of transactions is complete, then that request information shall be removed from the table. The Nested Request Table, Figure 3.6-1, shall be used in a similar manner for nested requests.

The RT Status Table, Figure 3.6-1, shall contain all the required status information for each RT. The bus controller shall have available to it all the information necessary to control complete bus operations. It shall also know what RT addresses are not being used at any particular time.

The Data Type/Priority Conversion Table, Figure 3.6-2, is utilized by the BC to convert data types to actual priorities so that transactions can be handled on a priority sequence basis.

Certain entries within the BC control tables are filled in by data at BC load time. Within the RT Status Table the following entries are loaded at initialization: RT in System, RT class and RT poll rate. Additionally, the entire Data Type/Priority Conversion Table is loaded at initialization.

These tables do not have to be implemented exactly as described but shall contain the minimum information necessary for the bus controller. More information and details can be included as the system requires.

BC CONTROL TABLES

Normal Request Table

REQUEST STATUS	ADDRESS OF REQUESTOR	ADDRESS OF ACCEPTOR	REQ T/R	CHAIN IND	SEQUENCE CODE	DATA TYPE	WORD COUNT

0 - No Req
1 - Req
2 - Trans Act

0 - Rec
1 - TR

0 - No Ch
1 - Poll Req - Ch
3 - Poll Acc - Ch

Nested Request Table

REQUEST STATUS	ADDRESS OF REQUESTOR	ADDRESS OF ACCEPTOR	REQ T/R	SEQUENCE CODE	DATA TYPE	WORD COUNT

RT Status Table

RT ADDR	RT IN SYSTEM	RT CLASS	POLL RATE	RT LINKED	OTHER RT ADDR	NEST ACTIVE	NESTED RT's ADDR	REQUESTOR OR ACCEPT	RT T/R	TRANS STATUS	RT STATU
Ø ↕ 3Ø											

0-OUT
I-IN

0-Polled
1-UnPolled

FIGURE 3.6-1 BC CONTROL TABLE

Data Type/Priority Conversion Table

Data Type	Priority Level
1 2 . . . 256	AS REQUIRED BY SYSTEM

Figure 3.6-2

DATA TYPE CONVERSION TABLE

3.7 RT Control Table

Each RT shall maintain a table of CDW's and TDW's as shown in Figure 3.7-1. Each RT shall utilize its Control Table to maintain and update its current status. If the RT was the requestor in an active transaction then the Request CDW shall reflect the state of the RT. If the RT was the acceptor of a request then the received Suggest CDW shall reflect the current state of that RT. The difference between the transmitted and received Suggest CDW's is determined by which RT is transmitting the data and which RT is receiving the data and not determined by which RT is the acceptor or requestor. The RT shall maintain a pointer to the area in the RT Control Table which is currently active. At any time the bus controller shall be able to extract as much of the table as desired by use of the sub-address (e.g., sub-address 14 with a count of 29 shall read the entire table). This function could be useful for error recovery and diagnostic purposes.

Unpolled RT's shall contain as much of the table as the CDW's and TDW's implemented by the RT. All RT's shall contain the last command word and last status word.

TYPICAL RT CONTROL TABLE

Corresponding SA	
14	NORMAL SUGGEST CDW (3) TRANSMITTED
15	NESTED SUGGEST CDW (3) TRANSMITTED
	NORMAL SUGGEST CDW (3) RECEIVED
	NESTED SUGGEST CDW (3) RECEIVED
16	NORMAL REQUEST CDW (3)
17	NESTED REQUEST CDW (3)
18	PROTOCOL COMMAND CDW (1)
19	LAST 1553B COMMAND
	LAST STATUS
30	TEST DATA WORDS (8 WDS)

Figure 3.7-1

RT Control Table

3.8 Protocol Sequences

The protocol sequences defined by this specification for P-3C MOD implementation are categorized as follows:

a) Mode Command Sequence (MCS)
b) Poll/Request Sequence (PRS)
c) Priority Service Sequence (PSS)
d) Transfer State Sequence (TSS)
e) Protocol Command Sequence (PCS)
f) Information Transfer Sequence (ITS)
g) Bus Test Sequence (BTS)
h) Broadcast Sequence (BS)

These sequences are defined and described in paragraphs 3.8.1 thru 3.8.5. Any sequence or variation from the above sequences not defined in these paragraphs shall be detected and the appropriate error status shall be generated.

3.8.1 Mode Command Sequence (MCS)

This protocol shall allow the use of all of the mode commands (except Dynamic Bus Control) defined by MIL-STD-1553B. The utilization of mode commands shall be device dependent except for specific uses defined in the error handling and test sections of this specification. The Mode Command Sequence can occur at any time that a particular terminal requires for correct operation. See Figure 3.8.1-1.

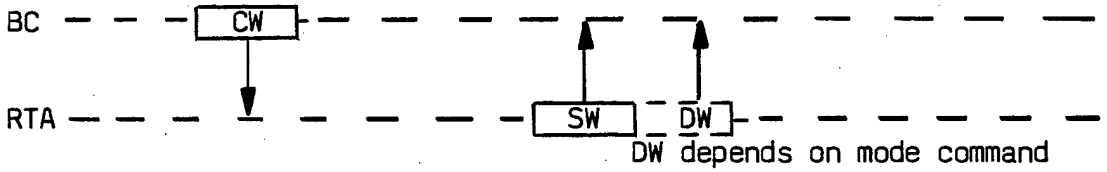


Figure 3.8.1-1

3.8.2 Protocol Control Sequences

The following sequences are variable length reads or writes between the BC and a RT. These reads and writes are addressed to the RT Control Table within the particular RT. The sub-address in the CW indicates the starting point of the read or write sequence and the word count indicates how many words are to be transferred.

The sequences described below are a few of the possible Protocol Control Sequences. These sequences shall be utilized throughout this document as examples of useful Protocol Sequences.

3.8.2.1 Basic Poll Request Sequence (PRS)

The Poll Request Sequence is a (transmit) command to an RT, to sub-address 16, and a word count of three. This causes a read of the RT's Normal Request CDW's as described in section 3.5.4. The basic Poll Sequence is shown in Figure 3.8.2.1-1 (a).

The purpose of the PRS is to permit each RT to input to the BC its bus requests, or needs. RT's are polled on a cyclic basis as determined by the RT's need to transfer data. The RT is always polled at this cycle rate independent of other bus traffic.

CW Format

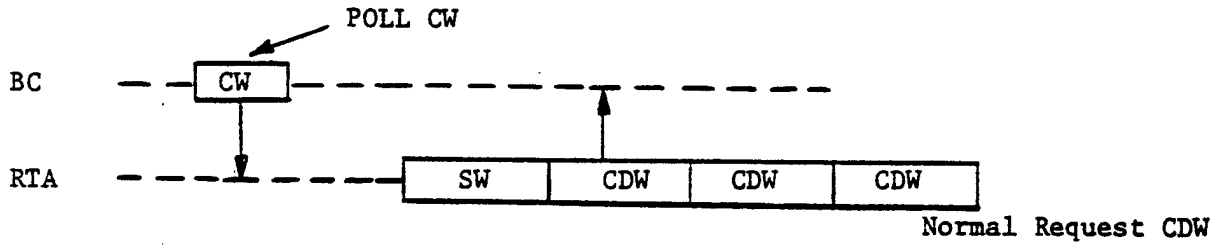
- Address is RTA
- T/R bit set to transmit (T/R = 1)
- SA is set to 16
- Word count is set to 3

CDW Format

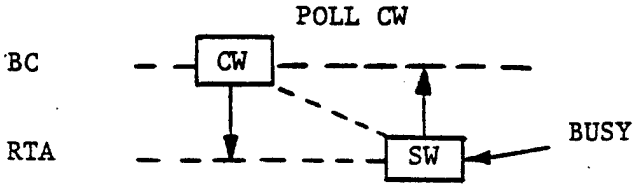
As defined in Section 3.5.5 for Request CDW

3.8.2.1.1 Busy Response to Poll Sequence

Figure 3.8.2.1-1 (b) shows the case where the response to a poll is a busy or condition. If busy, the poll is set to be repeated in the next minor cycle. This polling during each minor cycle continues until the RT is not busy. The protocol shall support this continuous polling but the application software shall determine how long this should continue on an RT by RT basis. The application software shall be capable of eliminating that RT from the poll, go into error recovery, alert the operator, or poll the RT at its normal poll rate.



(a) Basic Poll/Request Sequence



(b) Poll with Busy

POLL SEQUENCE

FIGURE 3.8.2.1-1

3.8.2.2 Priority Service Sequence

The Priority Service Sequence is described by Figure 3.8.2.2.-1. The BC uses this sequence to extract the Nested Request CDW's in response to an RT transmitting the Service Request (SR) bit in any of its Status Words. The SR indicates that the RT has a nested transaction to perform.

3.8.2.2.1 Sequence Format

3.8.2.2.1.1 CW Format

- o The address is RT
- o The T/R bit is set to transmit
- o The sub-address is set to 17 indicating transmit nested request
- o The DW count is set to three

3.8.2.2.1.2 SW Format

As defined in Section 3.5.3.

3.8.2.2.1.3 CDW Format

As defined in Section 3.5.5 for the Request CDW. The Sequence Code shall be 4, 5, or 6 and the Normal Data Word Count shall be from 1 to 2048.

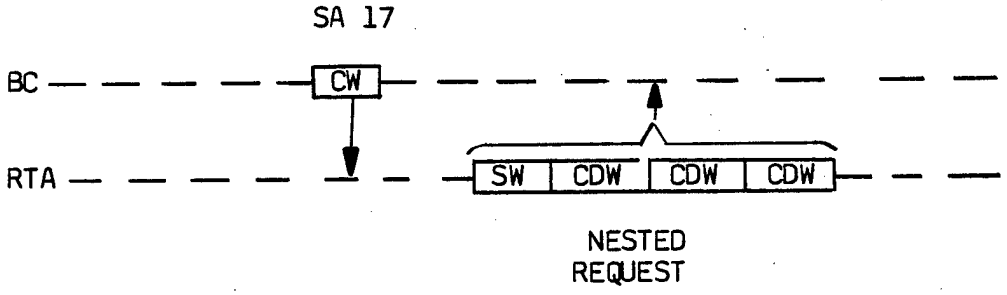


Figure 3.8.2.2-1

3.8.2.3 Transfer State Sequence (TSS) (Optional)

The Transfer State Sequence shall be used by the BC to modify the Transfer Code and Completion Status fields of a Normal Request CDW, Nested Request CDW, or the Suggest CDW contained in the RT's Control Table upon the completion of a transaction. The sequence shall always be used to update one of these fields when the BC detects an error. Specifically in an Rt-to-RT transfer the BC and the RT receiving the information know that the transaction has been completed, via the word count, but the transmitting RT needs an acknowledgement that the data was received by the other RT.

The TSS is a one word write to the RT Control Table of the particular RT. The write is to the location corresponding to the sub-address 16 for the Normal Request CDW, sub-address 17 for the Nested Request CDW, sub-address 14 for the Normal Suggest CDW, or sub-address 15 for the Nested Suggest CDW. Only the Transfer Code and Completion Status fields shall be updated when writing. The other fields shall be identical to the corresponding original CDW. The CDW being written to depends upon which terminal is transmitting the data. The BC always updates the RT Control Table of the RT transmitting the information, whether this RT is the requestor or acceptor. Since only the BC receives the the status words and both the BC and receiving RT shall be counting the data words, each knows when the transaction is complete. The transmitting RT shall inspect the Transfer Code and Completion Status fields after the TSS to determine the completion status of the transfer. If the transmitter was the requestor then the BC shall write to sub-address 16 or 17 for the Normal or Nested Request CDW. If the transmitter was the acceptor then the BC shall write to sub-address 14 or 15 for the Normal or Nested Suggest CDW. See Figure 3.8.2.3-1.

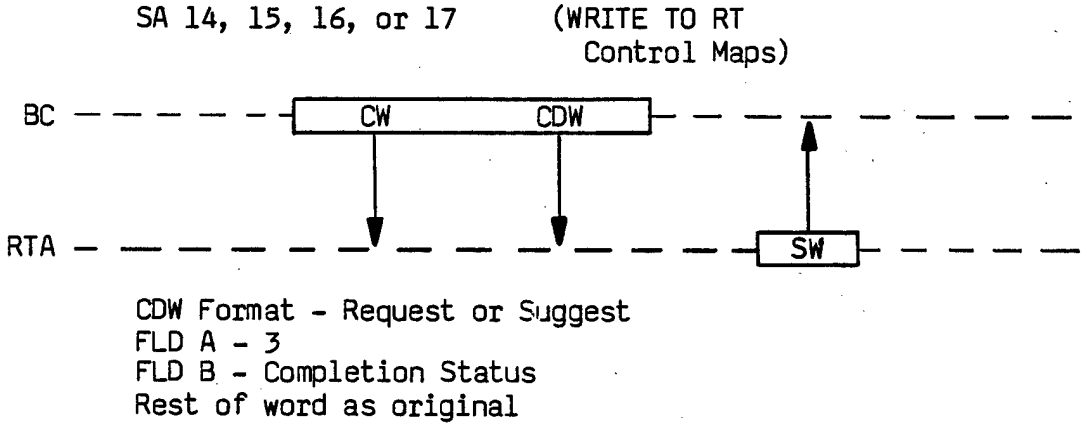


Figure 3.8.2.3-1

The intention of the TSS is a method of releasing resources (data buffers or processing) when a transaction has been completed. The specific resources are those within the transmitting RT. The TSS indicates to the transmitting RT that the sequence has been terminated. This sequence allows the BC to update only fields A and B. The balance of the word shall be appropriately filled in by the BC depending upon if the control word being updated is a Request CDW or a Suggest CDW.

If an RT's request is canceled due to an error condition then the TSS shall always be used to update Fields A and B of the Request CDW's to indicate the error conditions.

3.8.2.4 Protocol Command Sequence (PCS)

3.8.2.4.1 General

The purpose of this sequence is to provide the BC with complete control of the bus. The BC performs a write to the particular RT's Control Table, utilizing the Protocol Command CDW (sub-address 18). The RT shall then carry out the commanded operation.

A read PCS can be used, if needed for a particular RT, to determine if that RT has acknowledged or completed the command. (See Figure 3.8.2.4.1-1). The read PCS utilized in this manner shall take place during every minor cycle.

- Write PCS
  - CW Format
    - o Address is RTA
    - o T/R bit is set to receive
    - o SA is set to 18
    - o word count is set to 1
  - CDW Format
    - As defined in Section 3.5.7 for the Protocol Command CDW.
- Read PCS
  - CW Format
    - o Address is RTA
    - o T/R bit is transmit
    - o SA is set to 18
    - o Word count is set to 1
  - CDW Format
    - As defined in Section 3.5.7 for the Protocol Command CDW.

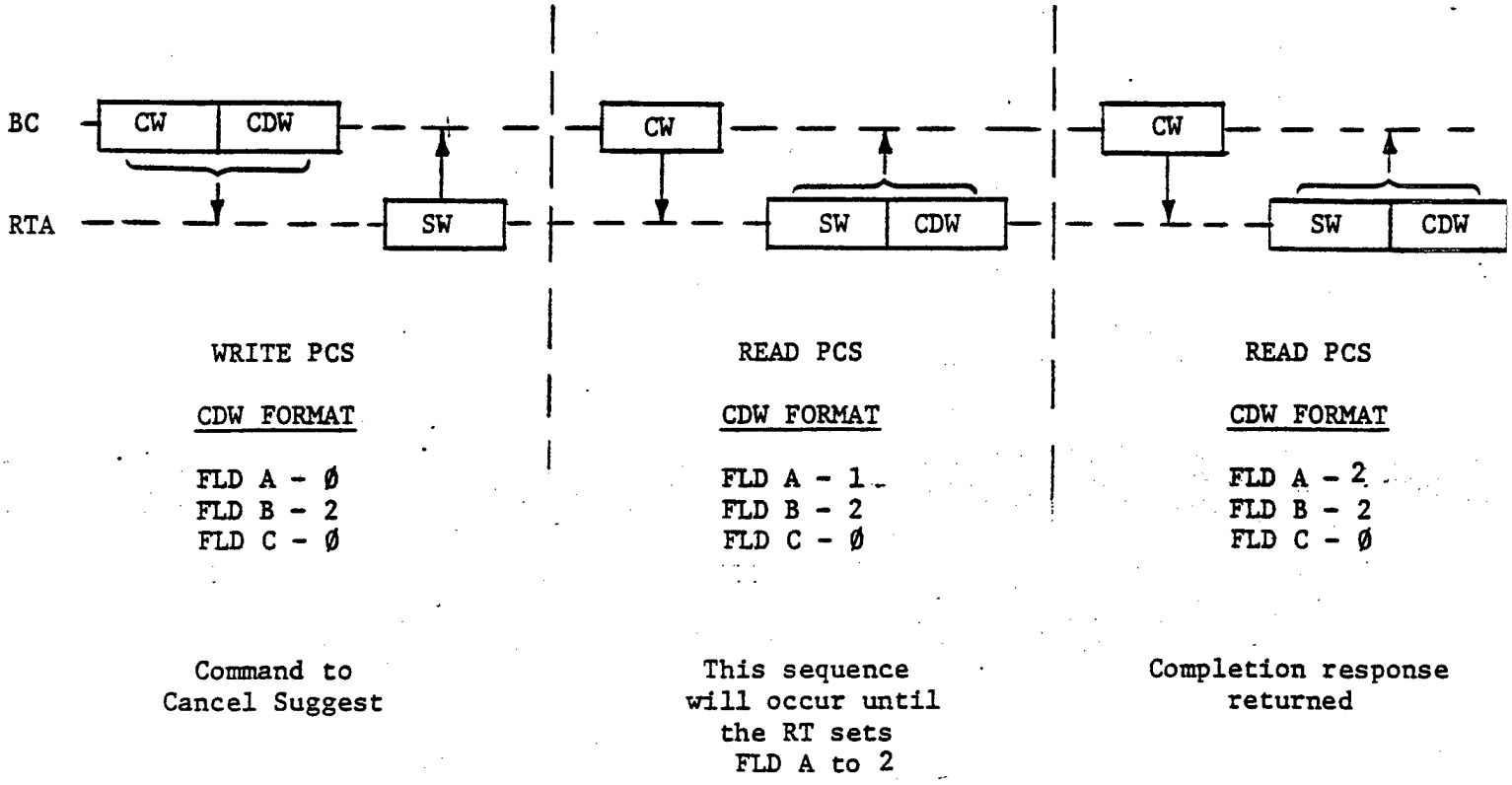


FIGURE 3.8.2.4.1-1 PROTOCOL COMMAND SEQUENCE

3.8.3 Information Transfer Sequence (ITS)

This is the only sequence which transfers actual application data words; that is, Interrupt Data Words (IDW) and Normal Data Words (NDW). The sequence codes in Figure 3.5.5.2-1 shall be used to define the transfer occuring in each block of the ITS. Each ITS consists of a single block or a multi-block transfer. The number of blocks in each ITS is a function of the word count in the Request CDW.

There are three types of Information Transfer Sequences:

ITS. 1

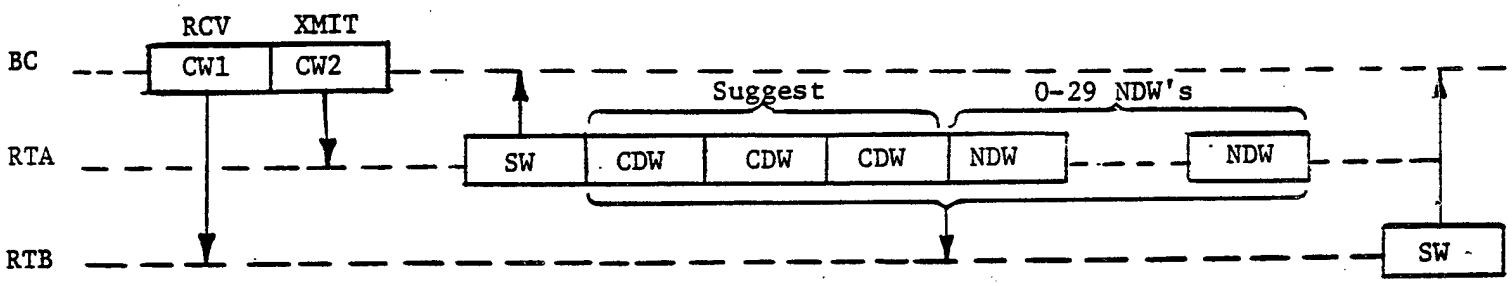
This is where the first three data words are Suggest CDW's and the remaining are NDW's. See Figure 3.8.3-1.

ITS. 2

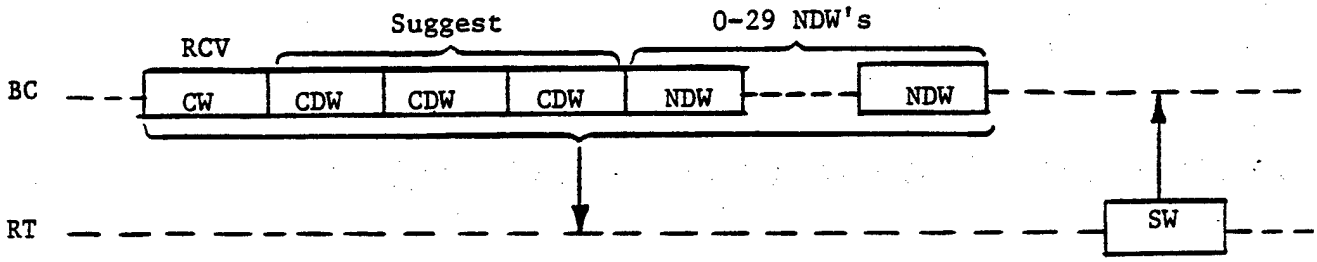
This is where the first three data words are Suggest CDW's, the next two are IDW's, and the rest of the block consists of from 0 to 27 NDW's. See Figure 3.8.3-2.

ITS. 3

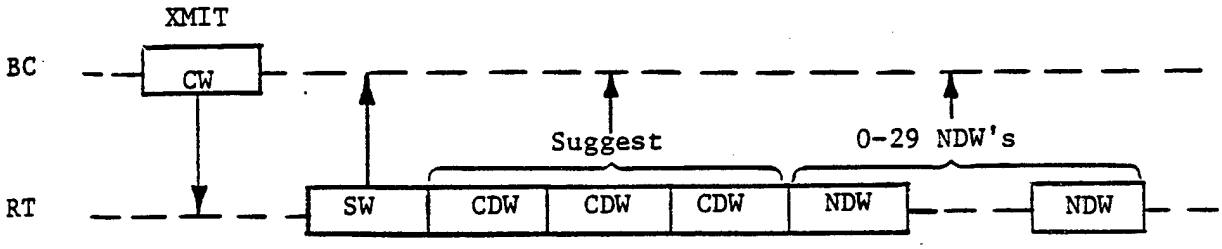
This is where all of the DW's in the block are NDW's. See Figure 3.8.3-3.



CW1, CW2: SA = 4, 20 or 24 (7, 23, or 27 for retry)
CDW: SEQ CODE = 4, 20 or 24
XFER CODE = 2
T/R = Ø (RTB to receive data)
(other fields of CDW encoded as required)
* 0 - 29 NDW's
(a) ITS.1 RT-to-RT XFER

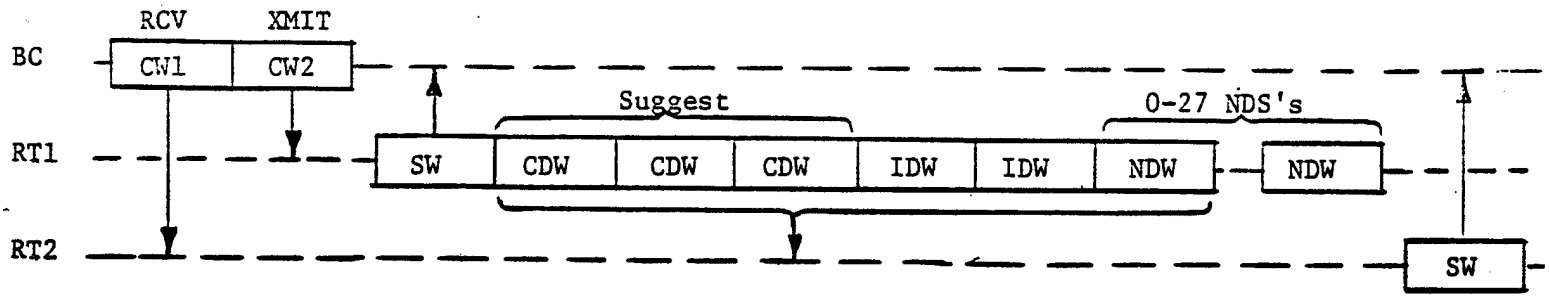


CW: SA = 4, 20 or 24 (7, 23 or 27 for retry)
CDW: SEQ CODE = 4, 20 or 24
XFER CODE = 2
T/R = Ø (RT to receive)
(other fields encoded as required)
(b) ITS.1 BC-to-RT XFER

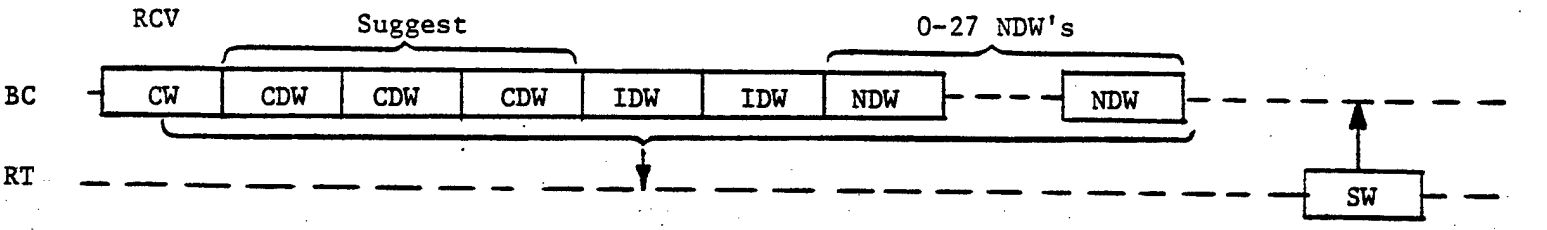


CW: SA = 4, 20 or 24 (7, 23 or 27 for retry)
CDW: SEQ CODE = 4, 20 or 24
XFER CODE = 2
T/R = Ø
(other fields encoded as required)
(c) RT-to-BC

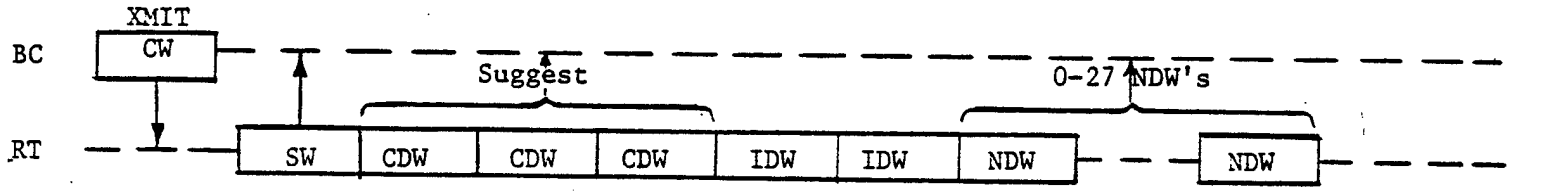
FIGURE 3.8.3-1 ITS.1



CW1, CW2: SA = 5, 21 or 25 (7, 23 or 27 for retry)
CDW: SEQ CODE 5, 21 or 25
XFEF CODE = 2
T/R = Ø
(other fields encoded as required)
(a) ITS.2 RT-to-RT

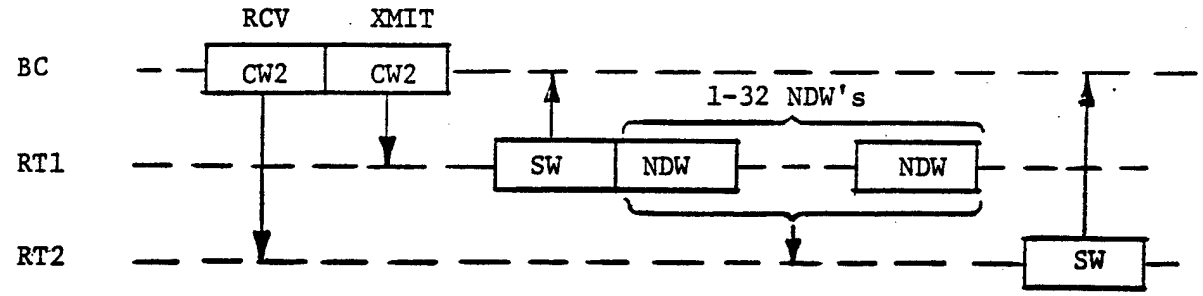


CW: SA = 5, 21 or 25 (7, 23 or 27 for retry)
CDW: SEQ CODE = 5, 21 or 25
XFER CODE = 2
T/R = Ø
(other fields encoded as required)
(b) ITS.2 BC-to-RT



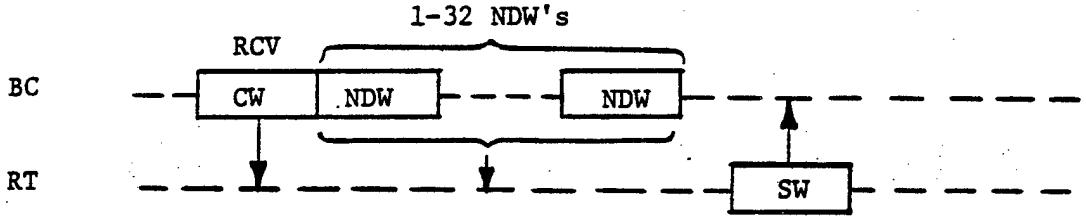
CW: SA = 5, 21 or 25 (7, 23 or 27 for retry)
CDW: SEQ CODE = 5, 21 or 25
XFER CODE = 2
T/R = Ø
(other fields encoded as required)
(c) ITS.2 RT-to-BC

FIGURE 3.8.3-2 ITS.2



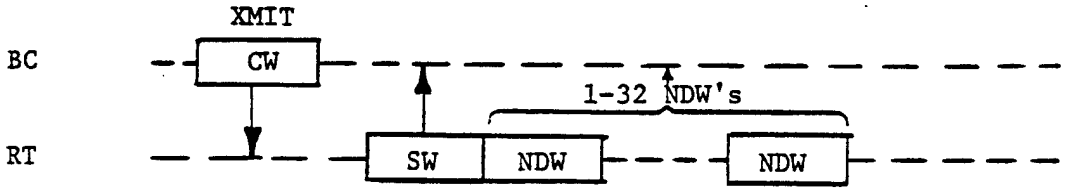
CW1, CW2: SA = 6, 22 or 26 (7, 23 or 27 for retry)

ITS.3 RT-to-RT



CW: SA = 6, 22 or 26 (7, 23 or 27 for retry)

ITS.3 BC-to-RT



CW: SA = 6, 22 or 26 (7, 23 or 27 for retry)

ITS.3 RT-to-BC

FIGURE 3.8.3-3 ITS.3

3.8.4 Test Sequences

3.8.4.1 Remote Terminal Self Test

The remote terminal shall, upon command, perform a remote terminal interface self test and present the results upon request. The command may either be a Mode Command or an application level command.

3.8.4.2 Bus Loop Test

3.8.4.2.1 General

The purpose of the Loop Test Sequence (LTS) is to determine if all data transfer sequences (BC to RT, RT to BC and RT to RT) are functional. The BC geneates all word formats and controls the sequencing of the test transfers. The RT only responds to the BC loading data to and from a fixed buffer area eight words in length with sub-address 30. The content of this buffer area is entirely controlled by the BC.

The test is partitioned into two sets of test data transfers. The first test, the BC to RT and RT to BC Loop Test, determines whether the BC can communicate with each RT unit and the second test, RT to RT Loop Test, determines whether each RT can both transmit and receive in the RT to RT mode. Both of these tests are run twice, the first time with the test pattern shown in Table 3.8.4.2.1-1, and the second time with the complement of the test pattern. The Manchester encoded TDW's are shown in Figure 3.8.4.2.1-1. The LTS types (A, B and C) that are utilized during the loop tests are defined in Figure 3.8.4.2.1-2.

3.8.4.2.2 BC to RT and RT to BC Loop Test

Figure 3.8.4.2.2-1 describes the BC to RT and RT to BC Loop Test. In this test the BC transmits the test pattern to each RT, receives the test pattern back from the RT, and compares the transmitted with the received data. This test is performed with each RT. The test pattern is then complemented and the test rerun.

This test is run before the test described in 3.8.4.2.3 in order that the complement of the test pattern is stored in the RT test buffers when the RT to RT Loop Test is performed.

3.8.4.2.3 RT to RT Loop Test

Figure 3.8.4.2.3-1 describes the RT to RT Loop Test. In this test the BC transmits the test pattern to the first RT. This is followed by an RT to RT transfer of the test pattern from the first RT to the second RT. The test pattern is then transmitted from the second RT to the BC where the received test pattern is compared against the transmitted pattern. This is repeated starting with the second RT to the next, and so on, until the last RT transmits to the first. The entire test is repeated with the complement to the test pattern.

Word	Test Pattern 4----------19				Parity 20
					Bit Time
0	1111	0000	1010	0101	1
1	0000	1110	0101	1010	0
2	1111	1111	1111	1111	1
3	0000	0000	0000	0001	0
4	1010	1010	1010	1011	0
5	0101	0101	0101	0101	1
6	1010	0101	1010	0100	0
7	0101	1010	0101	1010	1

TABLE 3.8.4.2.1-1 - TDW TEST PATTERN

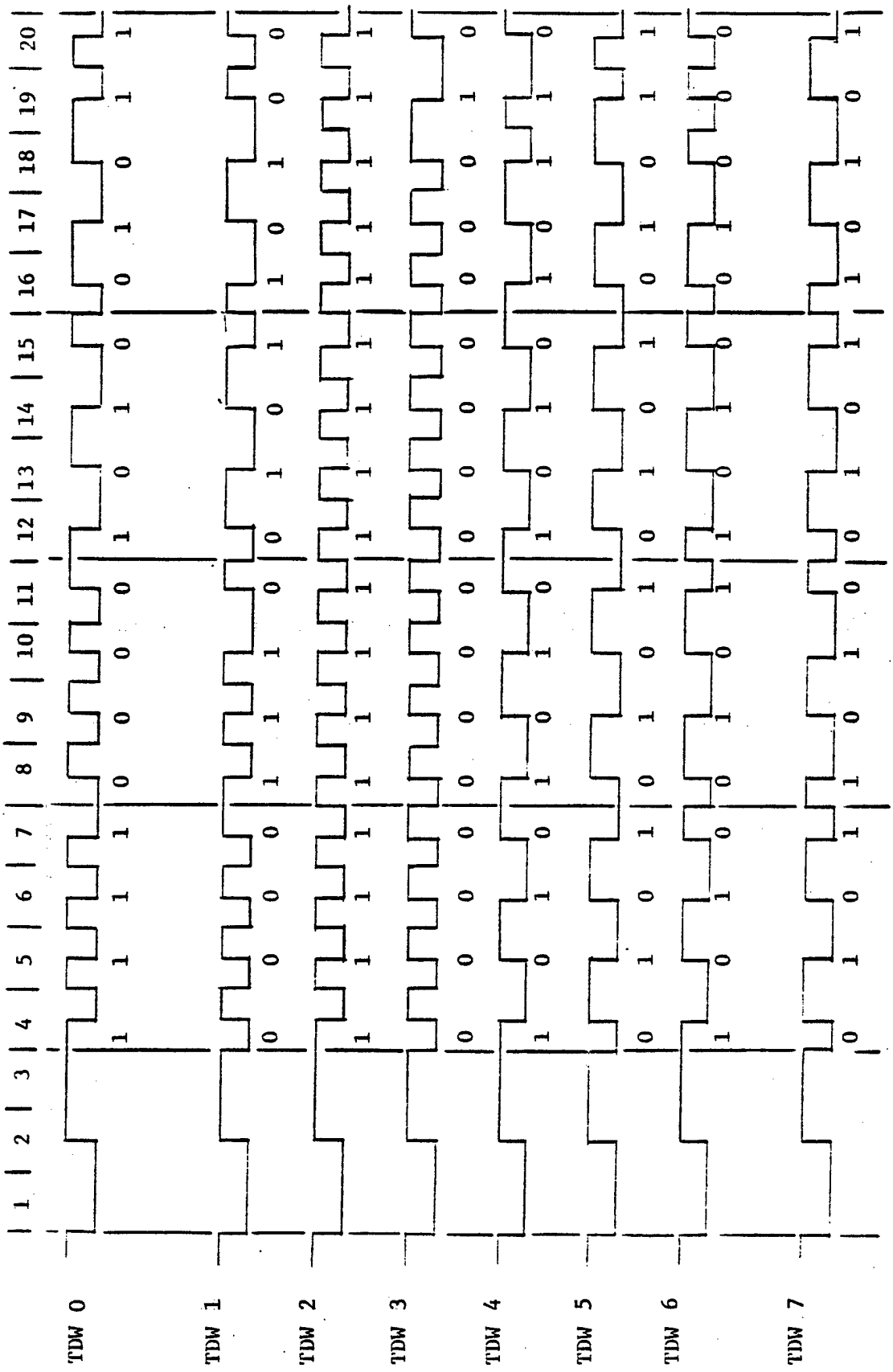
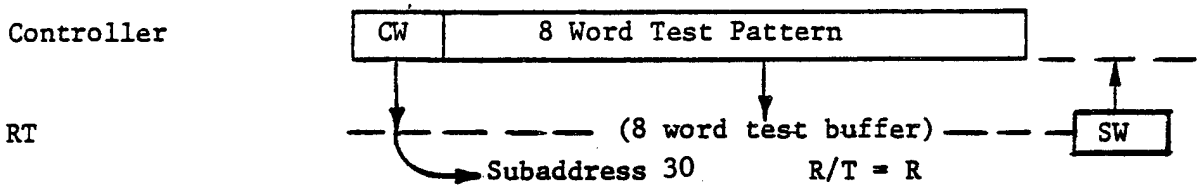
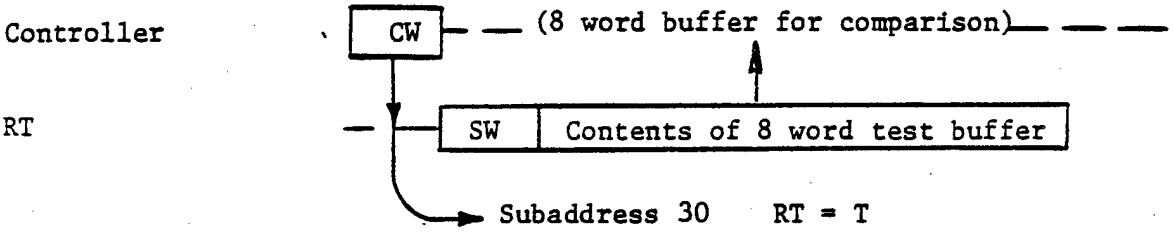


FIGURE 3.8.4.2.1-1 - MANCHESTER ENCODED TDW's

LTS Type A (BC to RT Transfer)



LTS Type B (RT to BC Transfer)



LTS Type C (RT to RT Transfer)

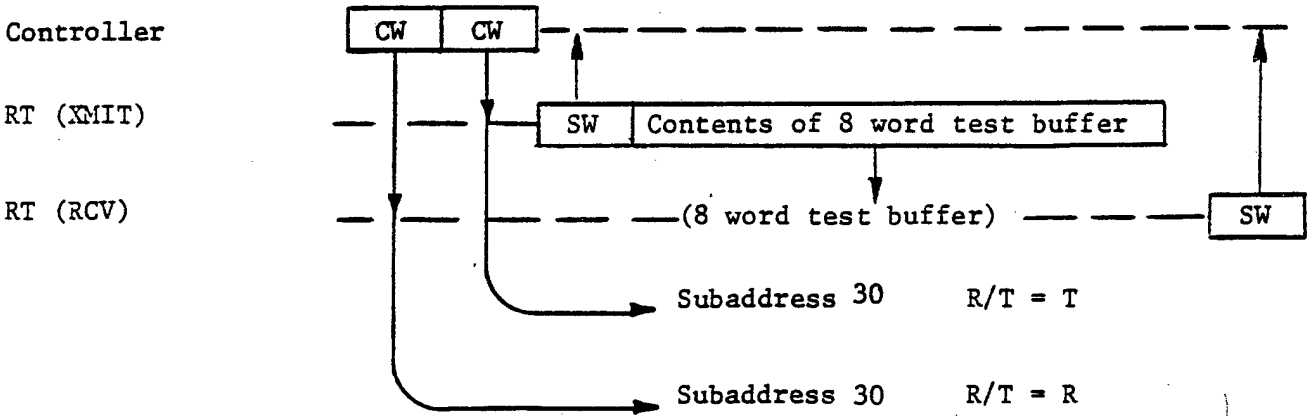


FIGURE 3.8.4.2.1-2 LOOP TEST SEQUENCES

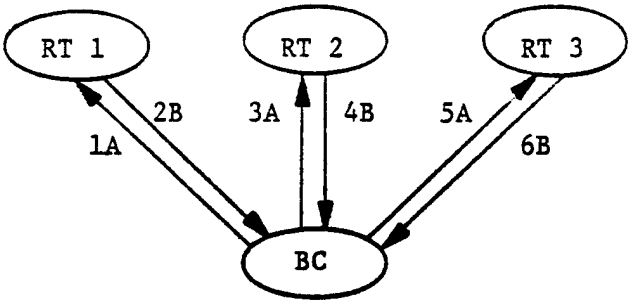
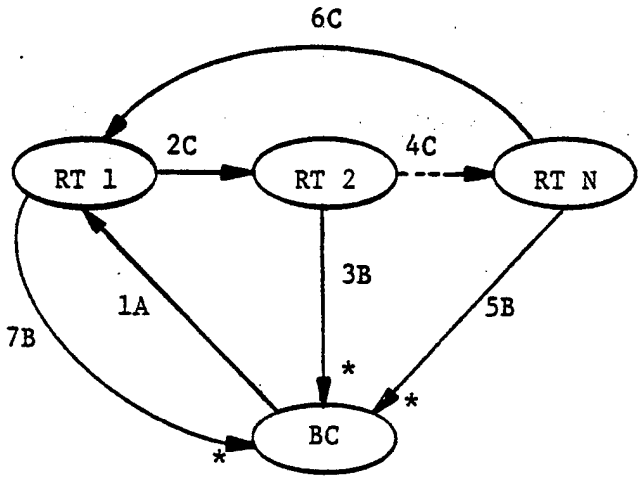


FIGURE 3.8.4.2.2-1

BC ⟶ RT AND RT ⟶ BC LOOP TEST



$N \leq 31$

A: BC ⟶ RT
B: RT ⟶ BC
C: RT ⟶ RT

FIGURE 3.8.4.2.3-1

RT ⟶ RT LOOP TEST

NOTES:

Numbers indicate sequence of events

Letters indicate type of transfer

*Indicates comparison of received data against transmitted data

3.8.4.3 Poll Test

3.8.4.3.1 General

The Poll Test is intended to verify that all polled RT's can respond to a Poll/Request Sequence. The BC shall transmit either a mode command Reset Remote Terminal (Mode Code 08), if the RT has implemented that mode code, or an application level RT reset followed by a Poll/Request Sequence (Sub-address 16). The BC shall verify that the Request CDW's returned are a legal response for that RT. This process shall be repeated for all polled RT's in the system.

3.8.5 Broadcast Sequence (BS)

This section describes as much of the Broadcast Sequence as this specification shall require. The remaining areas of this sequence and the specific utilization of the broadcast mode shall be further defined by the particular application and system implementation.

For the purpose of uniformity, there shall be certain restrictions in the use of the Broadcast Sequence which shall be adhered to by implementors. First, the sub-address field within the Command Word shall be defined exactly as in Figure 3.5.5.2-1. All other sub-addresses except the mode command sub-addresses 0 and 31, shall be illegal in Broadcast mode. Additionally, all Broadcast Sequences shall be limited to single block transfers. Multi-block transfers in broadcast mode presents a synchronization and data verification problem. The Bus Controller shall be responsible for verifying that all applicable RT's received the broadcast correctly. This verification shall be accomplished via a mode command requesting a status word, if implemented, or an application level transaction.

One possible use of the Broadcast Sequence is a broadcast utilized by selective groups of Remote Terminals (RT's). The definition of the groups of RT's is intentionally left to application level software, but all of the above restrictions shall apply (see Figure 3.8.5-1).

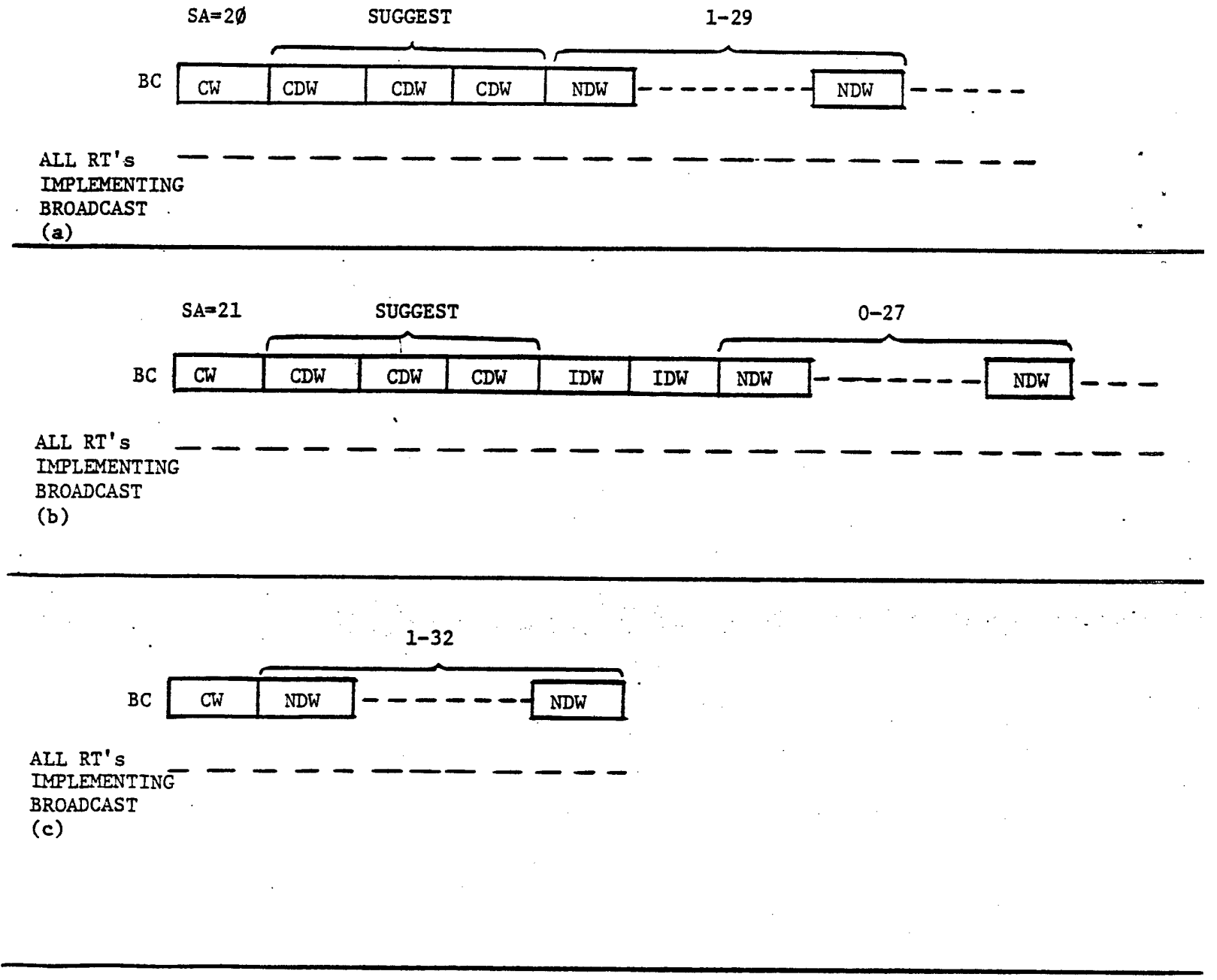


FIGURE 3.8.5-1

BROADCAST COMMAND TYPES

NOTE: Status of the Broadcast Command shall be determined via a transmit status word mode command or a transaction of application level information. This is for terminals implementing broadcast

3.9 Protocol Implementation

3.9.1 System Initialization

The normal system initialization sequence requires the operational software be loaded. As a result of this program load, the appropriate areas within the BC Control Tables shall be loaded (see Section 3.6). The BC software can now use the list of valid RT addresses to perform a bus initialization sequence.

3.9.1.1 Bus Initialization

The Bus Initialization function performs a quick check on the integrity of both Bus A and Bus B before the BC begins polling. The following functions are performed in the sequence indicated:

1. A mode command Reset Remote Terminal (Mode Code 08) is transmitted, if the RT has implemented that mode code, or an application level RT reset shall be sent to all RT's present on Bus B.

2. The Loop Test Sequence (see Section 3.8.4.2) shall be performed on Bus B.

3. A mode command Reset Remote Terminal (Model Code 08) shall be transmitted, if the RT has implemented that mode code, or an application level RT reset shall be sent to all RT's present on Bus A.

4. The Loop Test Sequence (see Section 3.8.4.2) shall be performed on Bus A.

If no errors are detected the BC shall be ready to begin polling on Bus A. If errors are detected the Bus Select program is called (see paragraph 3.9.1.2).

3.9.1.2 Bus Selection Function

3.9.1.2.1 Purpose

The purpose of the bus select function is to determine if a detected error is recoverable or permanent. The System Initialization routine will be used to perform this function.

3.9.1.2.2 Selection Criteria

The following procedure will determine which if any bus is to be selected when the Bus Select Function is called:

1. The bus initialization function is attempted three times on the present bus (e.g., Bus A).

2. If any of these three attempts of step 1 fail, the bus is immediately changed (e.g., to Bus B) and step 1 is attempted again.

3. If three consecutive initialization functions can be run on either bus, the BC resumes polling on that bus.

4. If the bus change occurs six times before polling is resummed the appropriate status is entered into the BC Control Tables, specifically in the RT Status Table, the field labeled RT Status, and the application software is notified of the error situation. A decision point has now been reached whether to go to a degraded mode, to get operator intervention, or to declare the bus as down. These decision processes are operational software dependent (Refer to Figure 3.9.4-1). The operational software may utilize protocol level tools in order to determine what path to take. If the decision is to continue in degraded mode the system initialization function may be re-entered with an updated list of valid RT's.

3.9.2 Transactions

3.9.2.1 General

A transaction consists of one of the following:

a) A Mode Command Sequence
b) Poll/Request Sequence, Information Transfer Sequence, and Optional Transfer State Sequence (Normal Transfer)
c) Priority Service Sequence, Information Transfer Sequence, and Optional Transfer State Sequence (Nested Transfer)
d) a single Protocol Command Sequence or a Protocol Command Sequence terminating a normal or nested transfer

Following are several examples of various types of transactions. A block diagram of each transaction is shown, followed immediately by the detailed implementation of that transaction. Each block of the diagram contains a three-letter abbreviation of the sequence that it represents and the arrows indicate the direction of 1553B data words. The detailed figures indicate what sequence is occurring at each stage of the transaction.

3.9.2.2 Single Block Unchained Transaction

A single block unchained transaction consists of a one ITS block transfers. There are three types of single block transfers allowable on the bus.

1) PRS, 1 ITS.1, an optional TSS
2) PRS, 1 ITS.2, an optional TSS
3) PRS, 1 ITS.3, an optional TSS

3.9.2.3 Multi-block Unchained Transaction

The multi-block unchained transaction consists of between 2 and 65 ITS block transfers. There are three types of multi-block transfers allowable on the bus.

1) PRS, 1 ITS.1, 1 to 64 ITS.3, optional TSS
2) PRS, 1 ITS.2, 1 to 64 ITS.3, optional TSS
3) PRS, 1-64 ITS.3, optional TSS

(See Figures 3.9.2.3-1 and 3.9.2.3-2 for an example of PRS, 1 ITS.1, 1 to 64 ITS.3, and an optional TSS transaction).

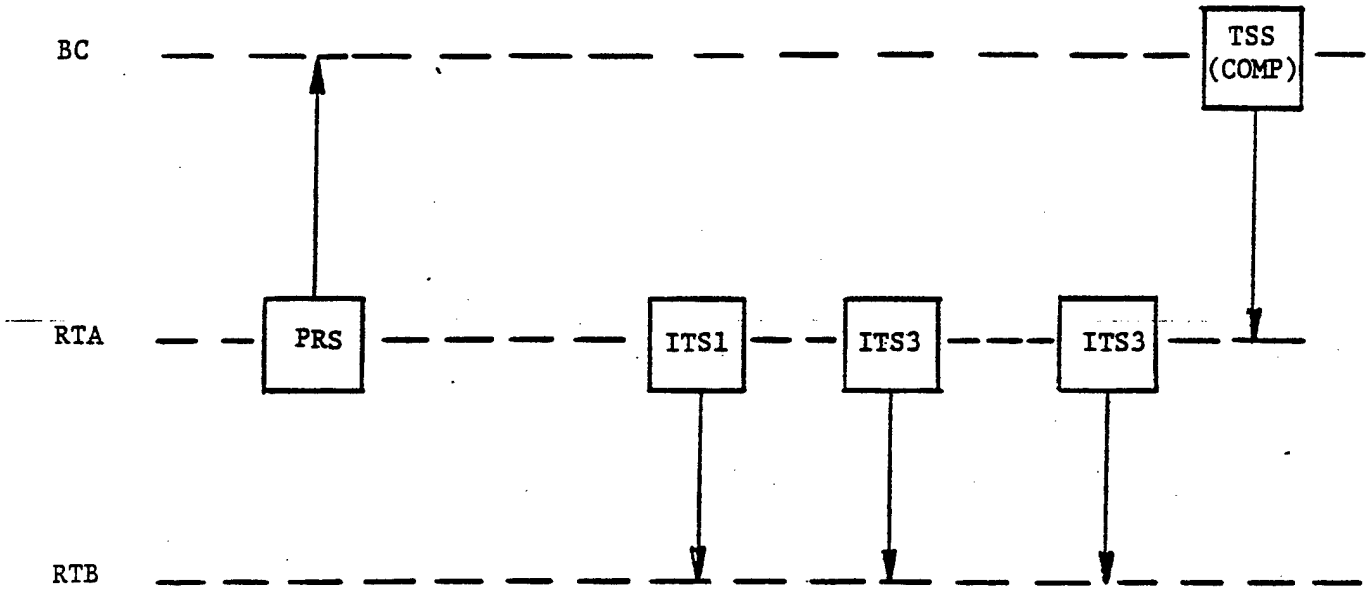


FIGURE 3.9.2.3-1

MULTI-BLOCK UNCHAINED TRANSACTION

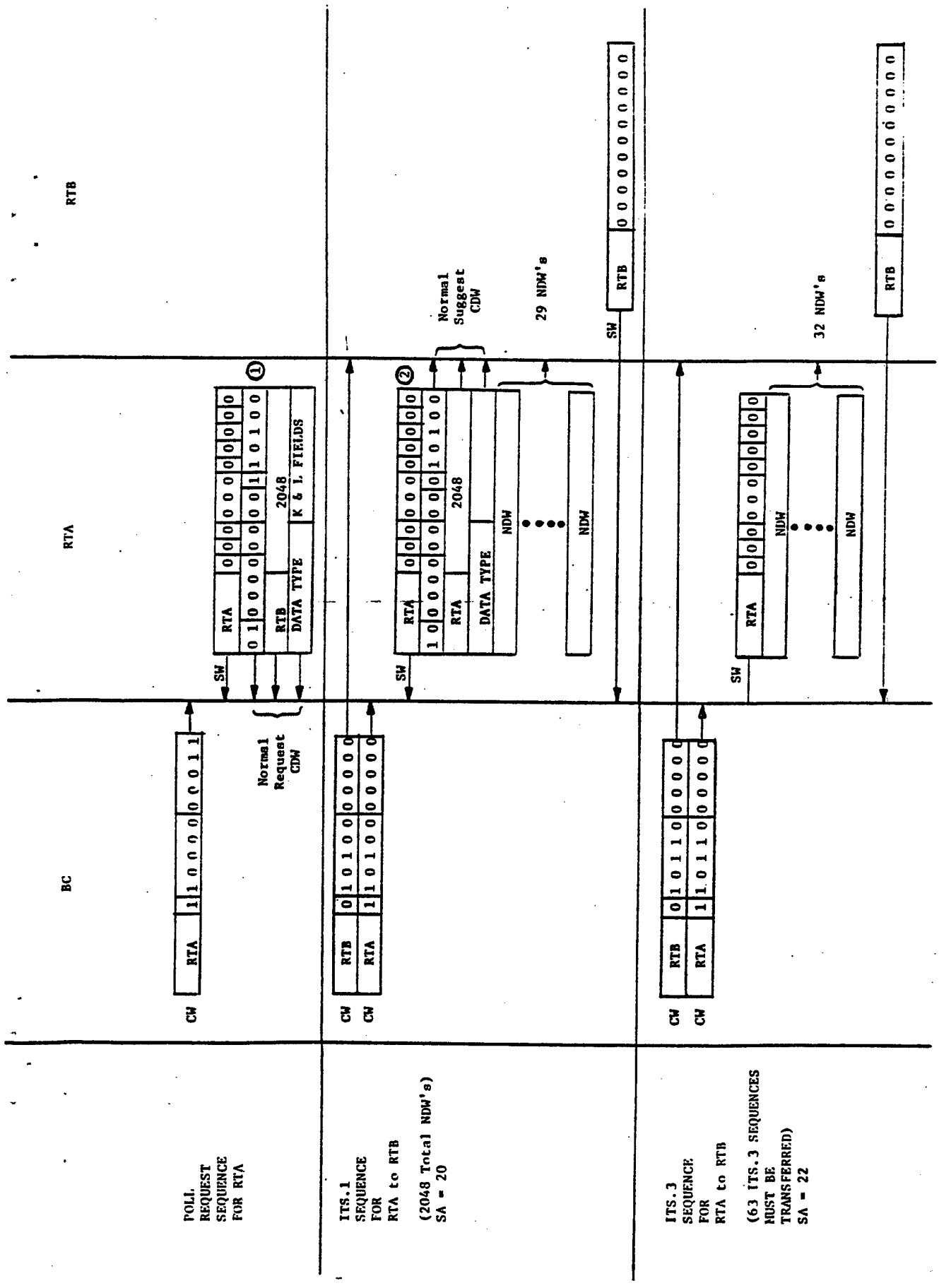


FIGURE 3.9.2.3-2 MULTI-BLOCK UNCHAINED TRANSACTION

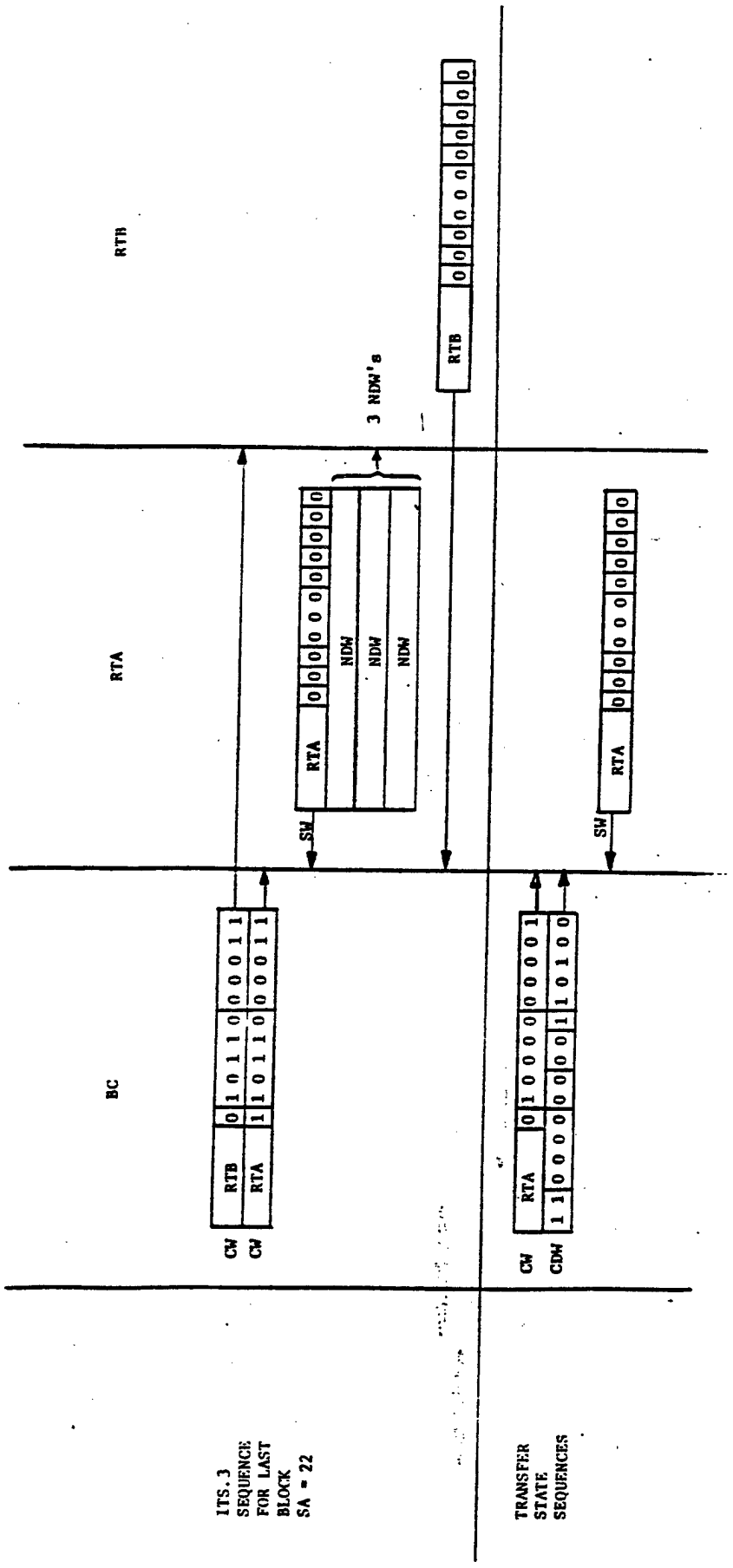


FIGURE 3.9.2.3-2 MULTI-BLOCK UNCHAINED TRANSACTION (cont'd.)

- RT-RT
- 2048 NDW's Transferred
- ITS.1 Sequence does not have to immediately follow PRS Sequence, sequences for other RT's can occur after this PRS but prior to this ITS.1
- RTB cannot Generate a normal request while linked to RTA

NOTES

1. Request CDW from RTA Control Table
2. Suggest CDW from RTA Control Table (Transmitted Suggest CDW) shall be stored in RTB's Control Table

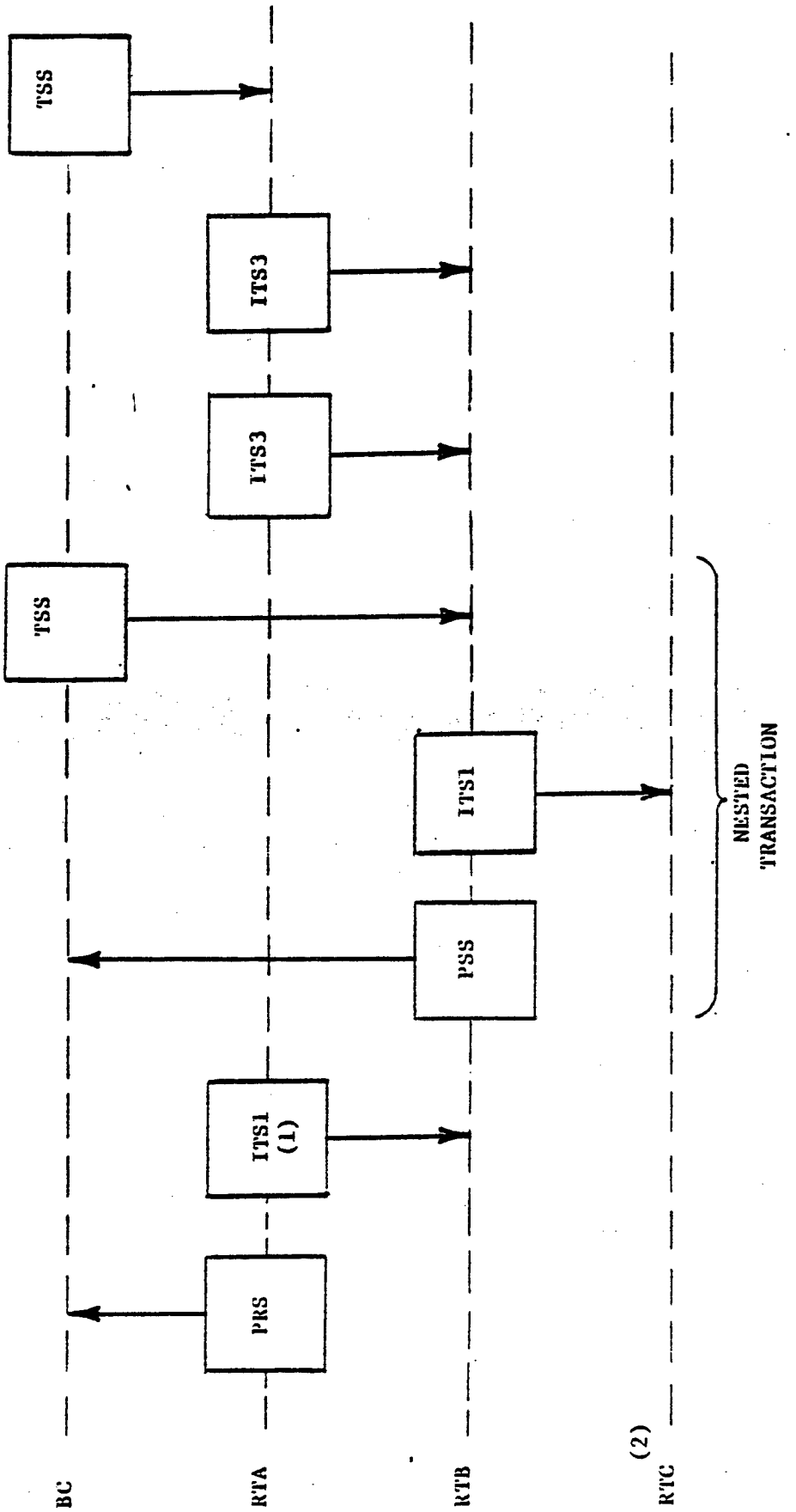
3.9.2.4 Nested Transaction

There are three types of nested transactions that this protocol defines.

In the first type, a linked RT has a requirement for a transfer of information with an unlinked RT. In this case, that RT shall flag the BC by setting the Service Request (SR) bit in its Status Word during a sequence of its current transaction. This informs the BC that this RT has a nested request. The BC shall perform a Priority Service Sequence (PSS) using sub-address 17 in the Command Word, to read the Nested Request CDW from the RT's Control Table. The Sequence Code field shall have a value of 4, 5, or 6. After reading the Request CDW, the BC shall compare the priority of the new request's data type with the priority of the active transaction's data type. If the new data type is higher priority, then the new request shall be executed before completing the old one. If the new data type is lower priority, then the BC shall use the Transfer State Sequence (sub-address 17 in the CW, Transfer Code of 3 and Completion Status of C, Insufficient Priority, in the CDW) to cancel the request. The RT can reschedule the request as required. Figures 3.9.2.4-1 and 3.9.2.4-2 illustrate this type of nesting.

In the second type, a linked RT has a requirement for a transfer of information with another linked RT. This type shall be executed exactly as the first except for two differences. The BC shall check that the priority of the new data type is higher than the priority of both linkages already active. If not, it shall terminate the request. The other difference is that the subaddresses in the CW's to the other RT are different from the first type of nesting. This is illustrated in Figures 3.9.2.4-3 and 3.9.2.4-4.

In the third type, an unlinked RT has a requirement for a transfer of information with a linked RT. In this case, the RT shall inform the BC of the request during the RT's poll cycle. The BC shall compare the priority of the new request's data type against the priority of the data type of the linked RT. If the new data type is of higher priority, then the requested transaction shall be executed before the completion of the old transaction. If the new data type is of lower priority, then the BC shall use the Transfer State Sequence (sub-address 16 in the CW, Transfer Code of 3, and Completion Status of C, Insufficient Priority, in the CDW) to cancel the request. The RT can then reschedule the request as required. Figures 3.9.2.4-5 and 3.9.2.4-6 illustrate this type of nesting.



(1) Status word from RTB has SR Bit Set
(2) RTC is not linked

FIGURE 3.9.2.4-1 NESTED TRANSACTION TYPE 1

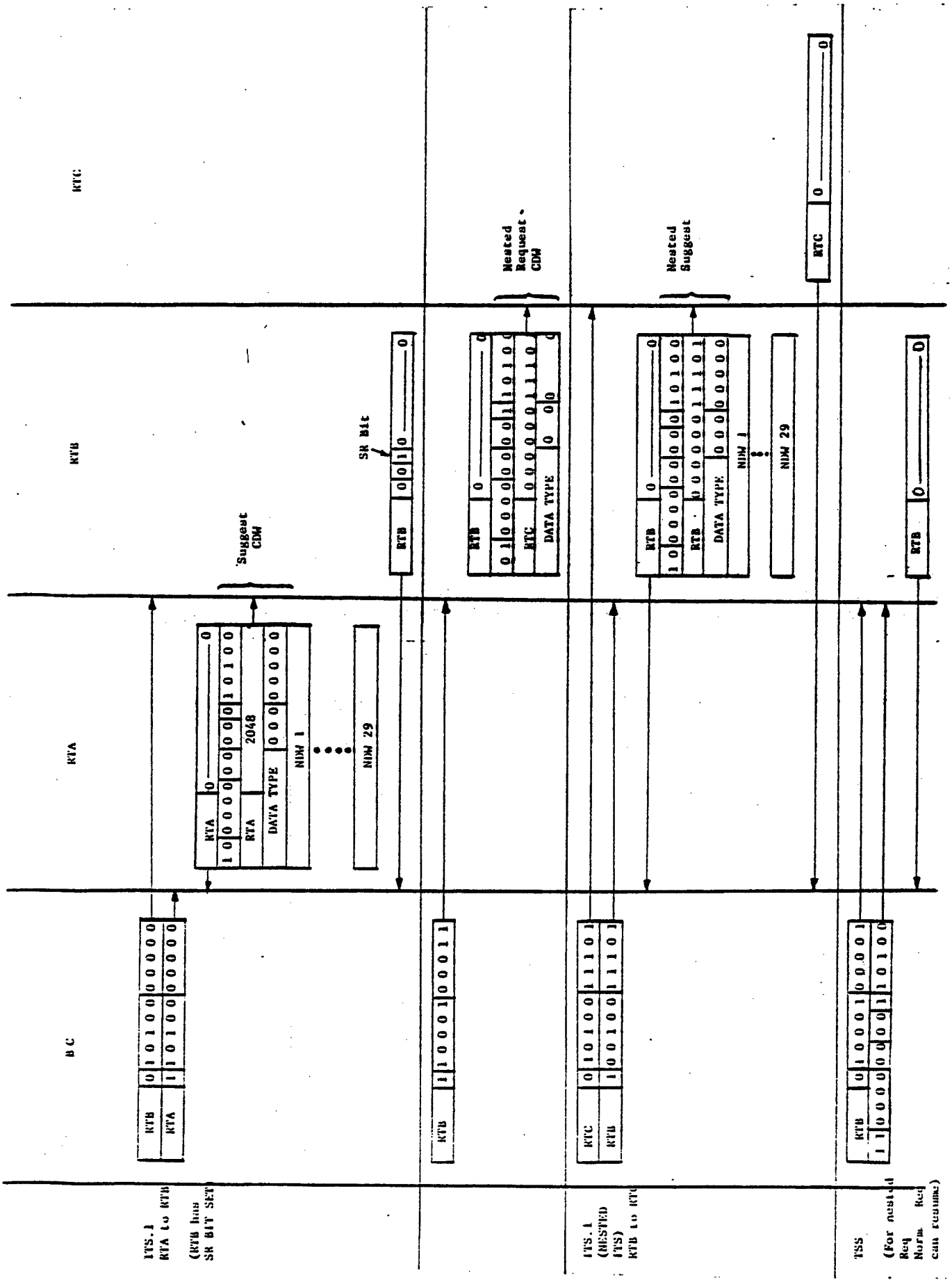
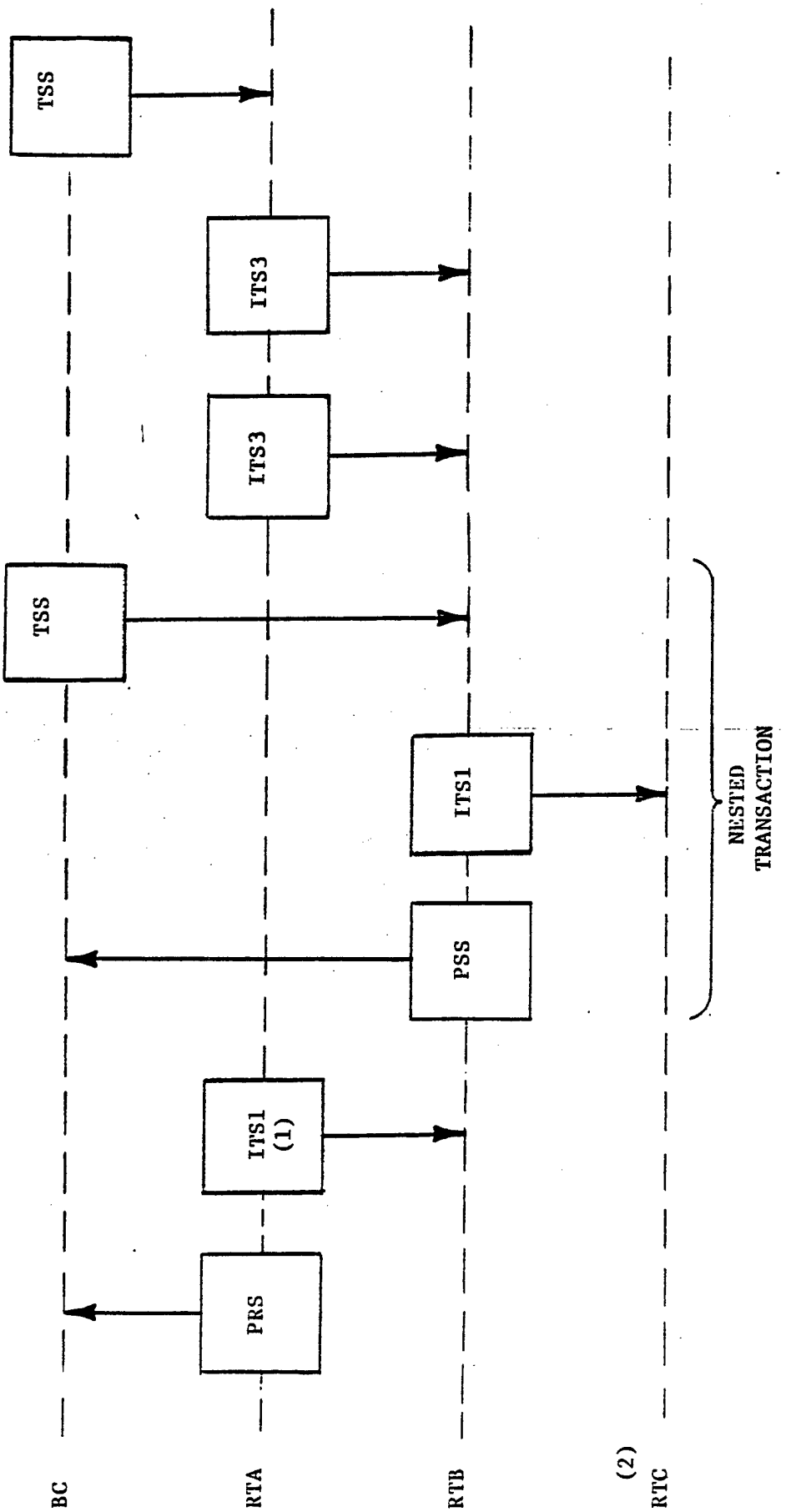


FIGURE 3.9.2.4-2 NESTED REQUEST TYPE 1



(1) Status word from RTB has SR Bit Set
(2) RTC is linked

FIGURE 3.9.2.4-3 NESTED TRANSACTION TYPE 2

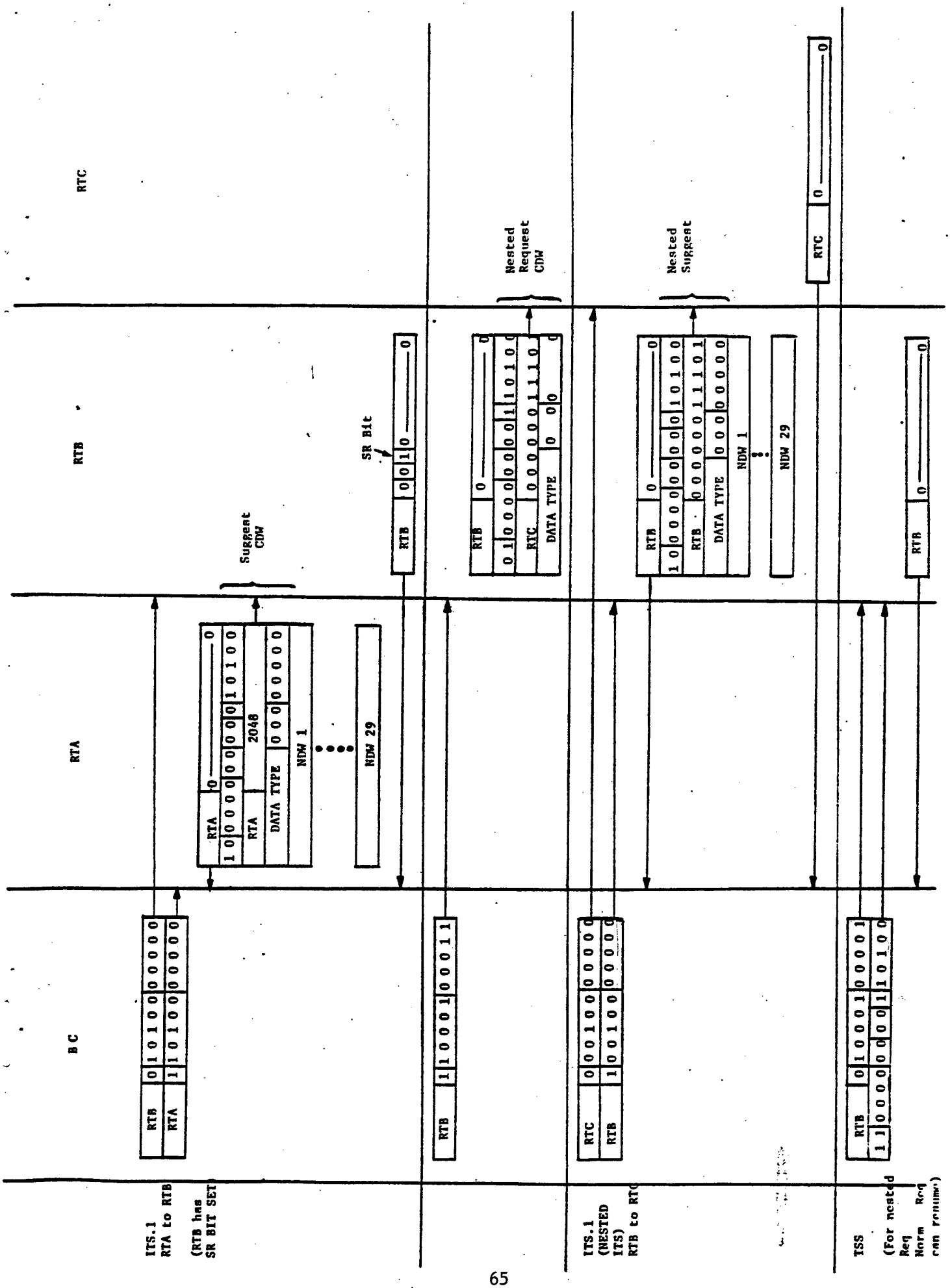
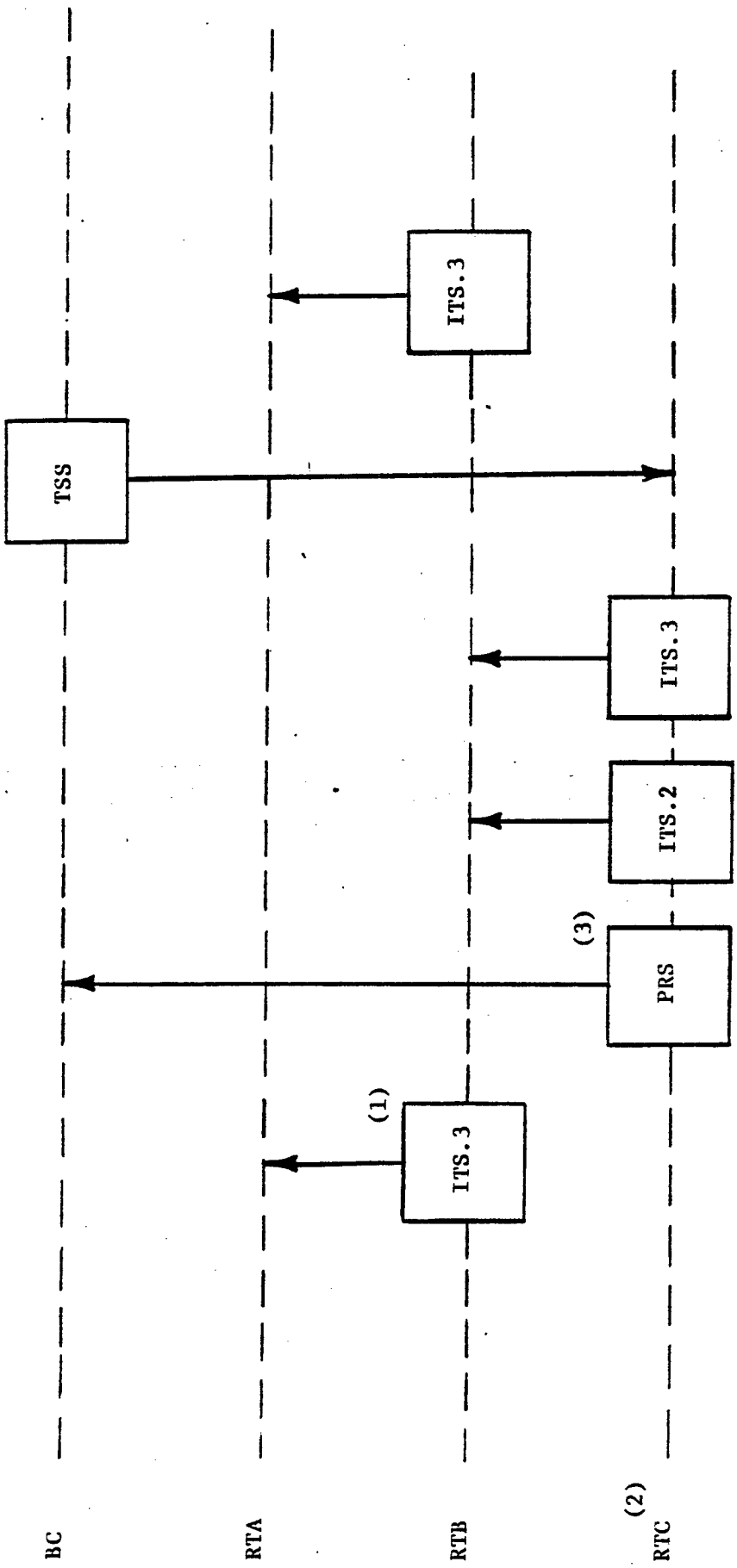


FIGURE 3.9.2.4-4 NESTED REQUEST TYPE 2



(1) An unchained multi-block transaction is active between RTA and RTB
(2) RTC is not linked
(3) This is the poll cycle for RTC

FIGURE 3.9.2.4-5 NESTED TRANSACTION TYPE 3

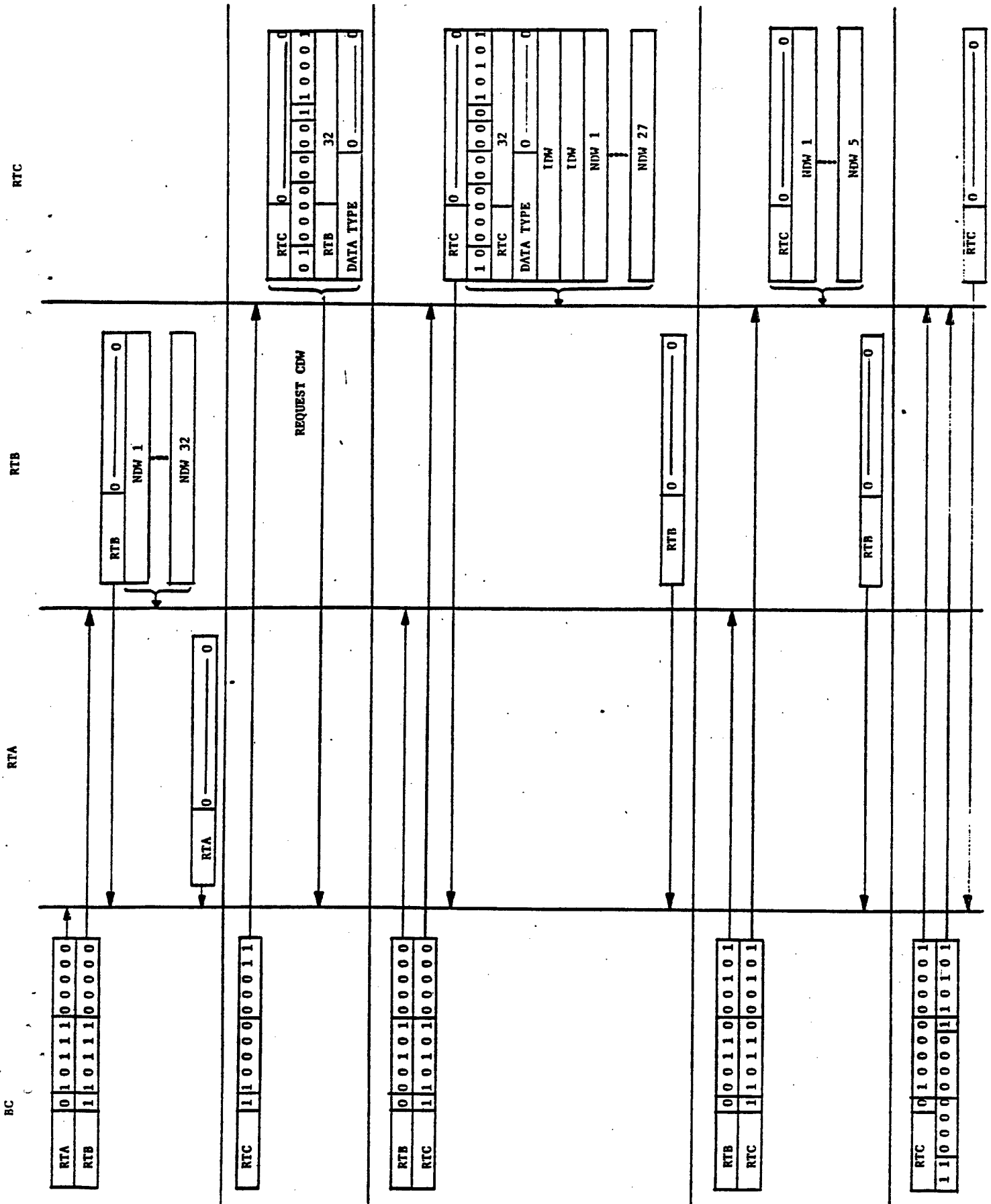


FIGURE 3.9.2.4-6 NESTED TRANSACTION TYPE 1

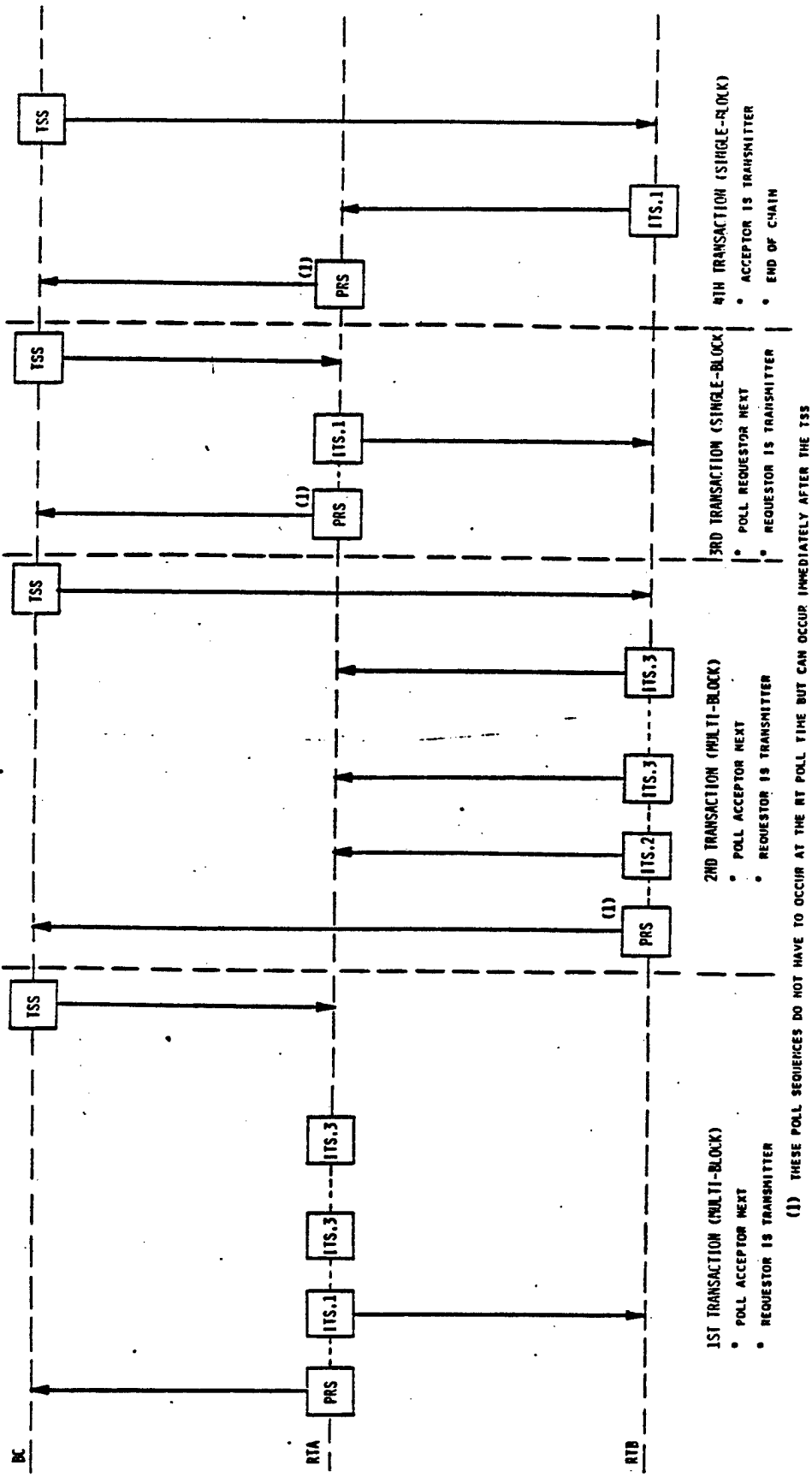
3.9.2.5 Chained Transactions

A chain of transactions is one in which during the poll cycle of the first transaction, the requestor tells the BC one of two things to do when the transaction is complete: poll requestor next or poll acceptor next. This shall occur for the poll sequence of each transaction, except for the last transaction which shall tell the BC that it is the end of the chain. The chaining action is controlled by Field D, the Chaining Indicator field, of the Request CDW. Chaining requires only the first transaction to contain a Suggest CDW. Chaining is meant to be used between two RT's which have predefined series of transfers to peform between them, in any combination of directions, which can continue with minimal overhead control. Each succeeding transaction of the chain can occur at the earliest time that the BC can schedule it. That is, the poll sequence for the next transaction shall occur at the earliest time the BC can schedule it but no later than that RT's cycle time. If the next request of the chain is not ready, the process shall continue until the request is ready.

A chain of transactions can consist of any combination of transactions except that the first transaction shall contain an ITS.1 or ITS.2. That is, a Suggest CDW shall be the first thing transferred to the other RT.

An example of chaining is shown in the block diagram, Figure 3.9.2.5-1, and the detailed implementation of this diagram, Figure 3.9.2.5-2.

This example shows how two Remote Terminals can transfer a predefined series of transactions utilizing the chaining function available in this protocol. These two RT's have a few different chaining sequences available between them (for this example we are using chaining sequence identifier #2). The first transaction transfers 200 normal data words from RTA to RTB. The second transaction transfers 2 interrupt data words followed by 50 normal data words from RTB to RTA. The third transaction is a single block transaction which transfers 12 normal data words from RTA to RTB. The last transaction of the chain is a single block transaction which transfers 9 normal data words from RTB to RTA.



(1) THESE POLL SEQUENCES DO NOT HAVE TO OCCUR AT THE RT POLL TIME BUT CAN OCCUR IMMEDIATELY AFTER THE TSS OR AT THE EARLIEST TIME THE BC CAN SCHEDULE IT AND NO LATER THAN THE RT'S NEXT POLL CYCLE.

FIGURE 5.9.2.5-1
EXAMPLE OF CHAINED TRANSACTION

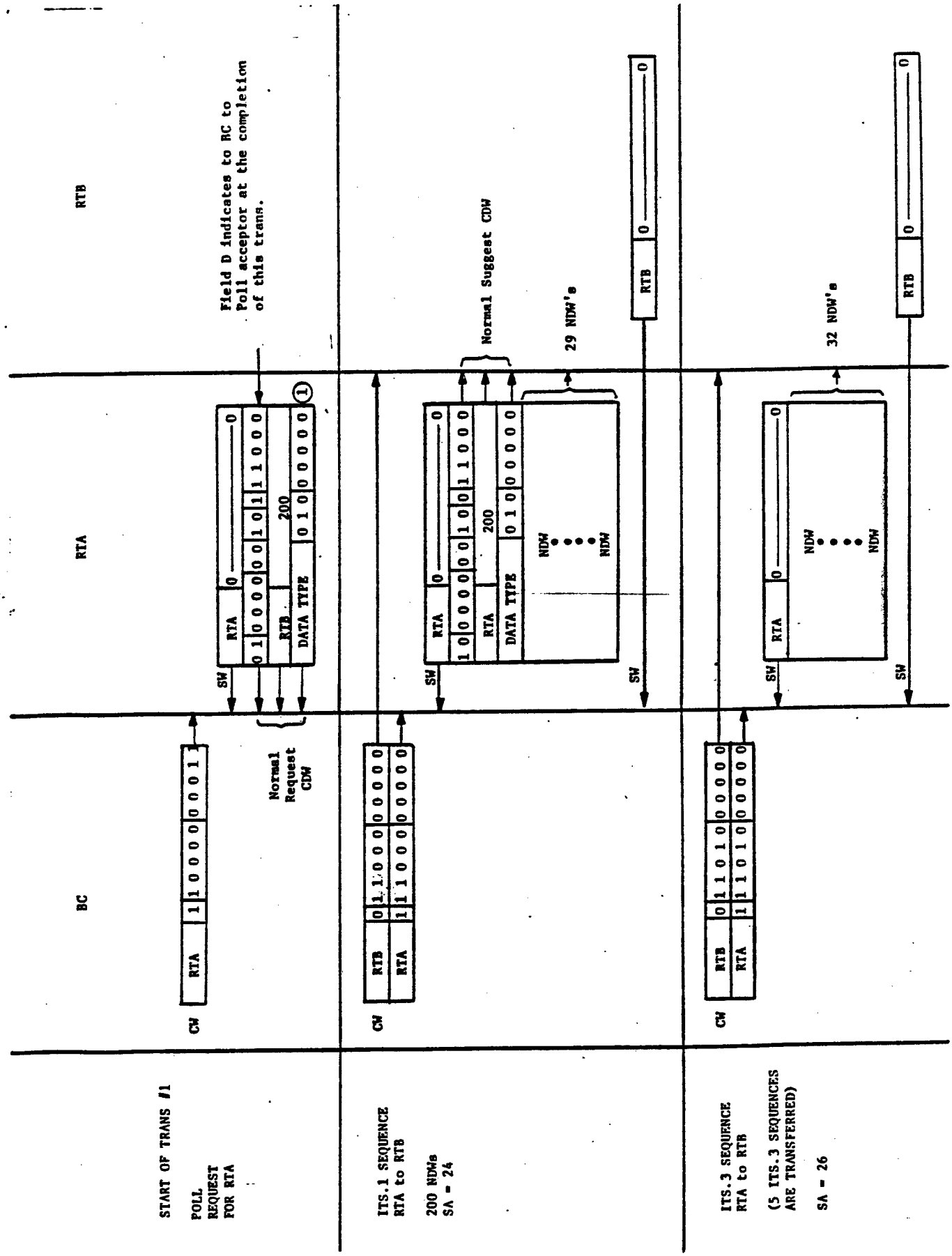


FIGURE 3.9.2.5-2 EXAMPLE OF CHAINING

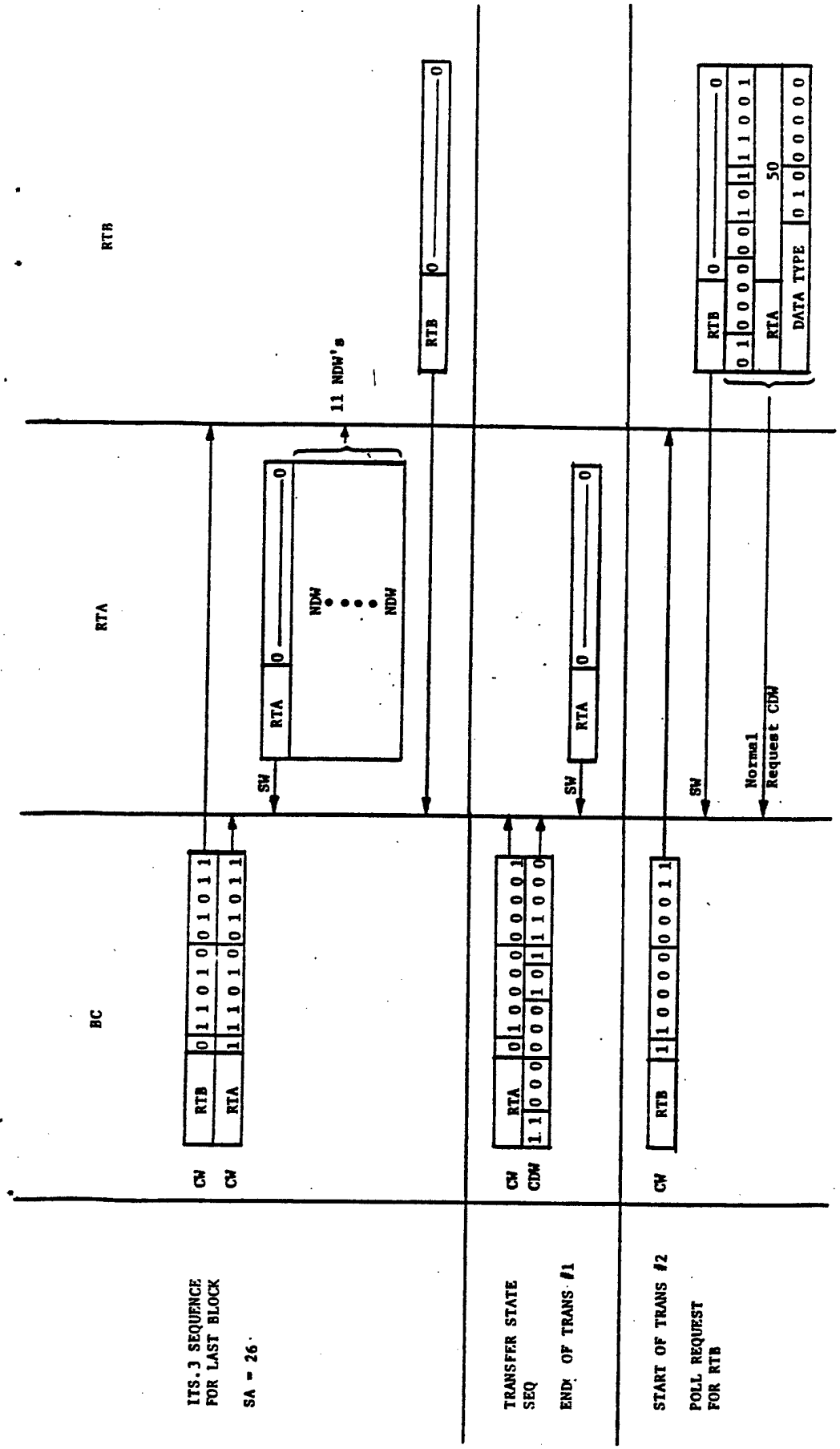


FIGURE 3.9.2.5-2 EXAMPLE OF CHAINING (cont'd.)

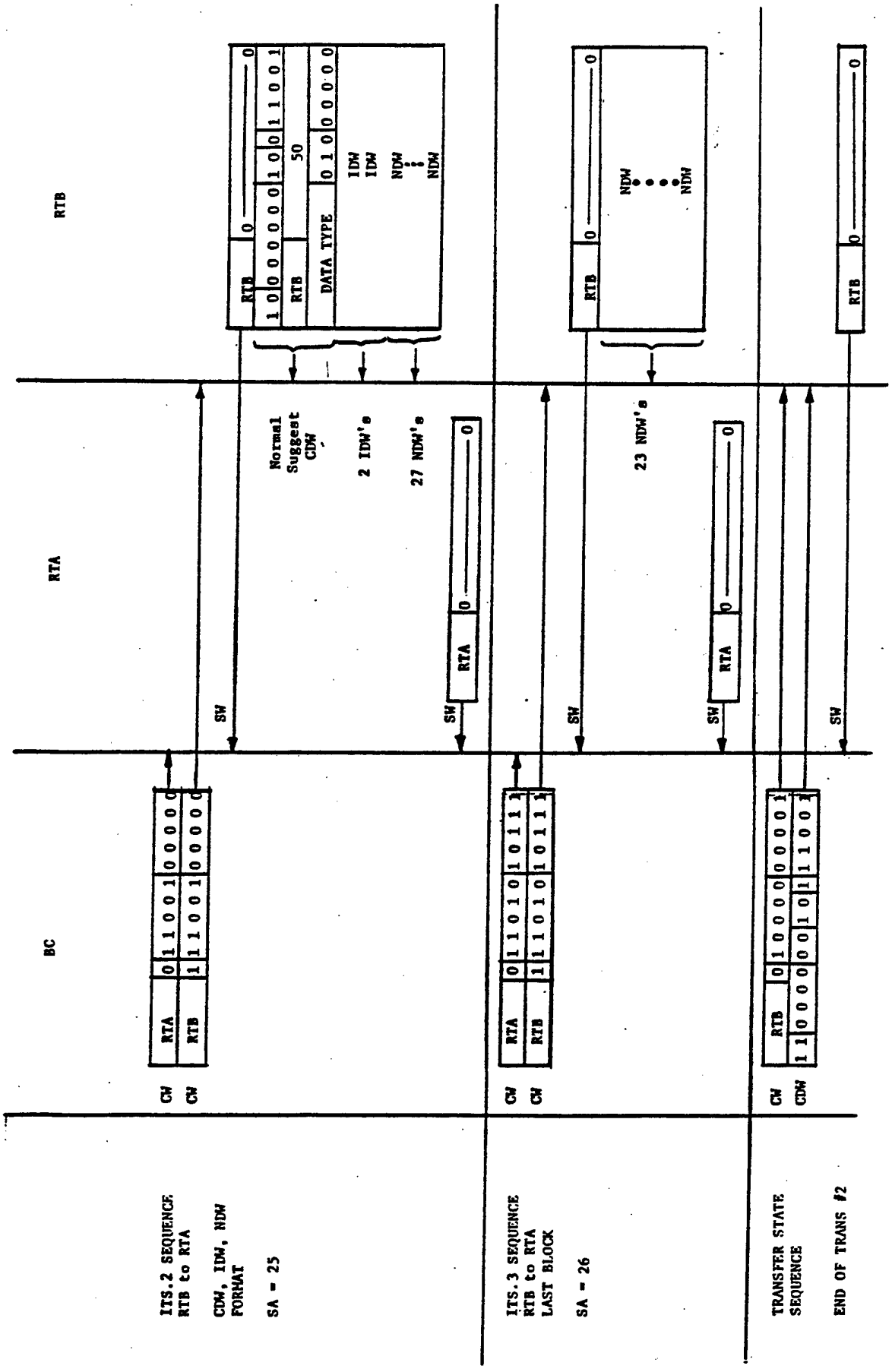


FIGURE 3.9.2.5-2 EXAMPLE OF CHAINING (cont'd.)

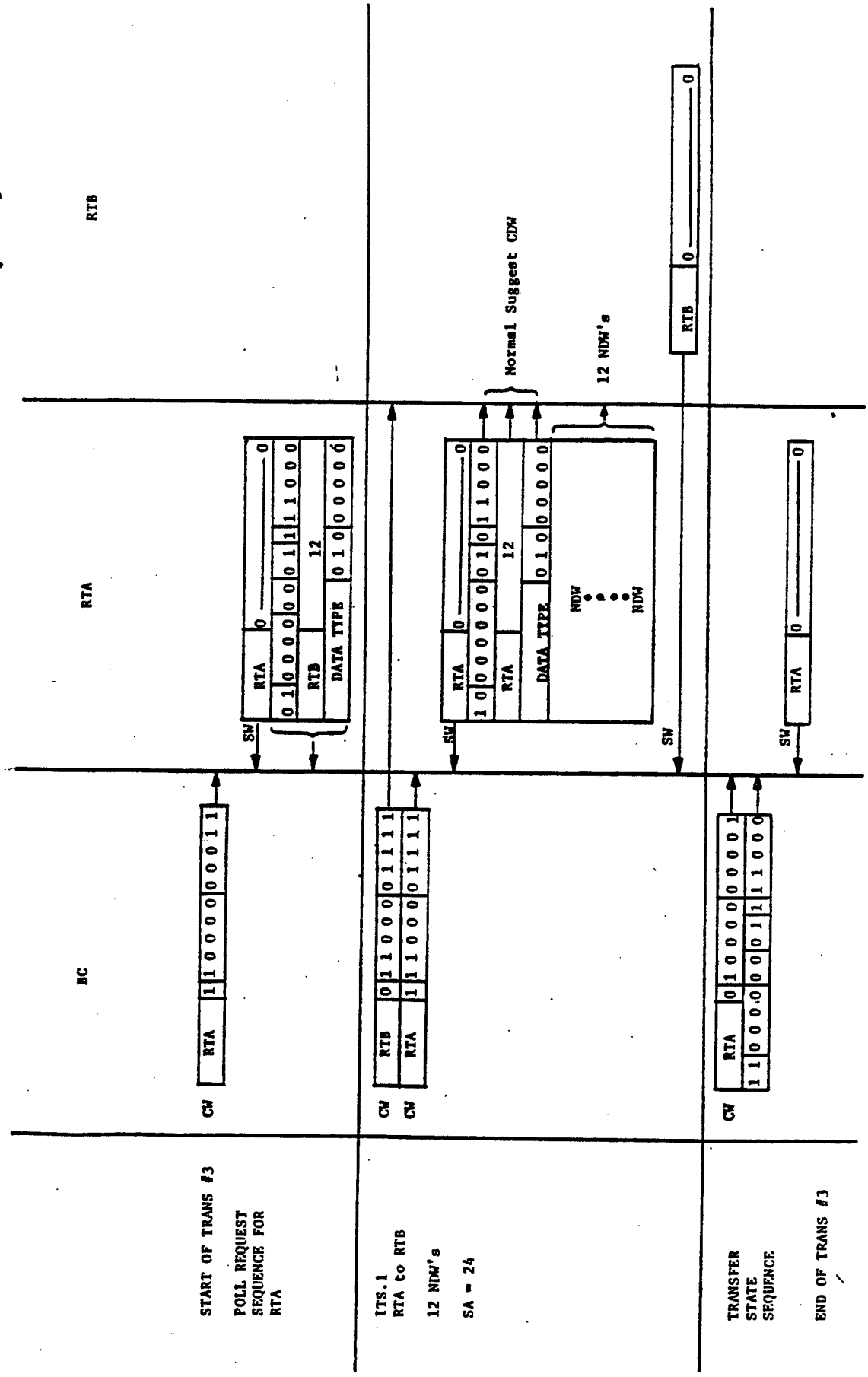


FIGURE 3.9.2.5-2 EXAMPLE OF CHAINING (cont'd)

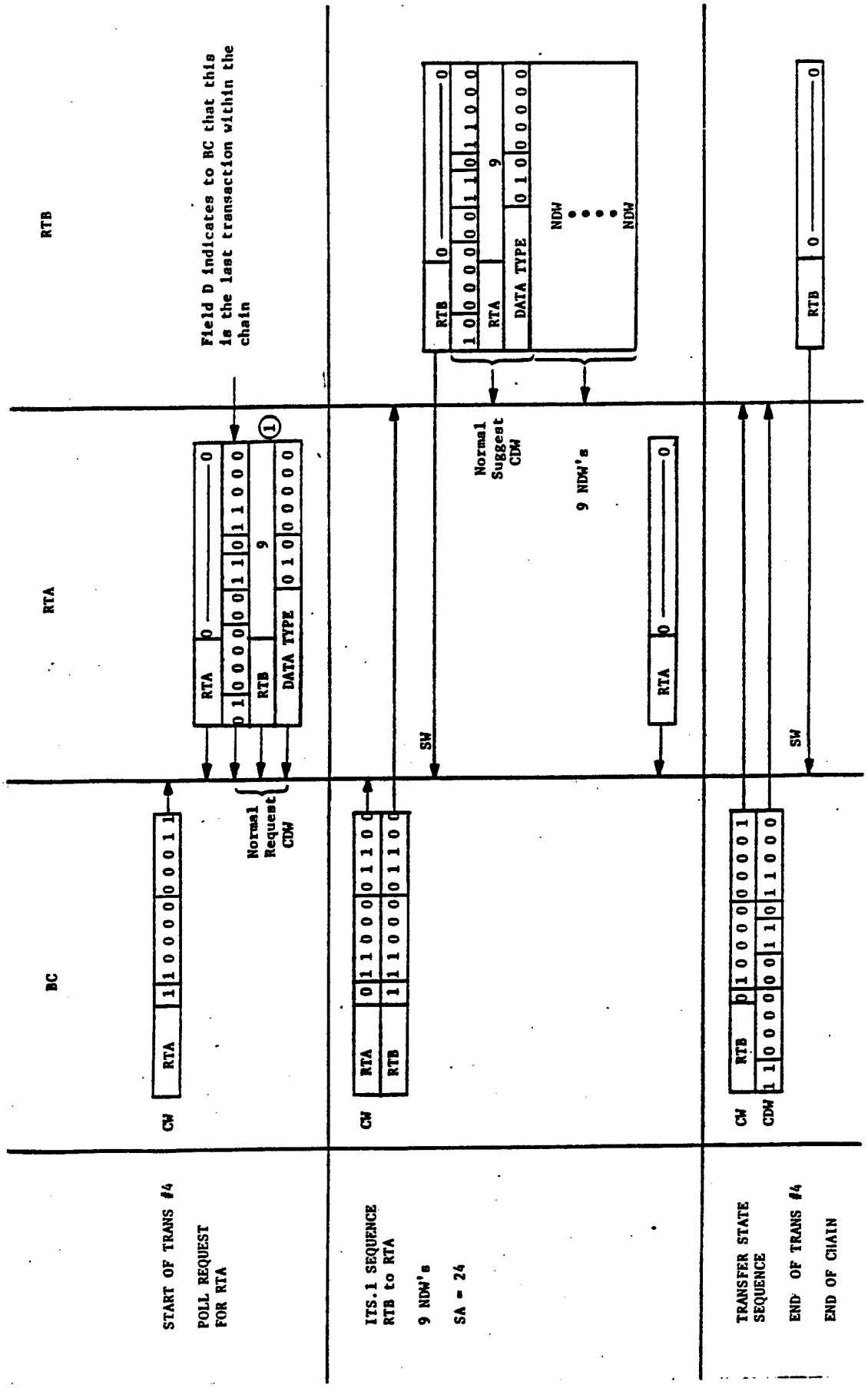


FIGURE 3.9.2.5-2 EXAMPLE OF CHAINING (cont'd.)

3.9.3 Automatic Test Procedures

The Bus Loop Test has at least three identified uses. Firstly, this test shall automatically periodically be run in the operational environment (on a non-interfering basis) as an in-flight-performance-monitoring test. This background testing shall include alternating the busses in order to determine bus readiness. Results of this testing shall be noted within the BC Control Tables, specifically in the RT Status Table, the field labeled RT Status. Bus status shall also be available.

The Bus Loop Test is intended to be utilized during initialization and diagnostic testing. It may be used either one at a time, as discussed above, or in the Continuous Bus Loop Test mode.

3.9.3.1 Continuous Bus Loop Test

3.9.3.1.1 Purpose

The purpose of the Continuous Bus Loop Test is to determine bus error conditions over an extended period of time (minutes to hours). The BC runs this test by continually rerunning the Loop Test Sequence. Application level software intervention is required in order to invoke the Continuous Loop Test Sequence. Any error detected by the BC during any type of transfer with any RT is classified as transmit or receive relative to the RT. A receive and transmit error counter is required for each RT. An additional counter is required to count the number of times the LTS is repeated.

3.9.3.1.2 Procedure

The test requires a keyset, display and operator. The keyset is required to initialize and terminate the test and the display is required for an operator to evaluate the error counters associated with each RT. The following functions are required to run the test:

1. Operator selects Bus Integrity Test
2. Operator selects Bus A or B
3. Operator initiates test

Loop tests run continually increments 2n+1 counters as required

4. Operator stops test and evaluates number of tests and the transmit error counter and receive error counter for each RT.

3.9.4 Error Recovery Function

The error recovery program provides a means of recovering from simpler transient errors by retrying sequences. If a simple retry will not correct the problem the Bus Select function is called. All error recovery tools are shown in Figure 3.9.4-1.

3.9.4.1 Priority

Error recovery will have the highest priority in the BC sequencing algorithm. When an error is detected the function in which the error was detected is immediately retryed.

3.9.4.2 Error Retry Function

The error retry function shall be the first method of error handling performed by the BC. This function shall be utilized upon the detection of an error in an RT Status Word (e.g., Message Error, RT Fault, Subsystem Fault). The transmission shall be repeated up to two more times if the error continues to occur. If the error disappears normal processing shall continue, if not the Bus Select Function shall be called.

There are certain errors which cause no SW response from the RT. In this case, the BC shall extract the SW by implementing a mode command Transmit Status Word (Mode Code 02). This SW should help the BC to determine what caused the no response. The error retry function can now continue.

In the case of an error the BC shall transmit the identical command as it was originally transmitted, except if the command was transmitted during an Information Transfer Sequence. In this case the sub-address of the command shall be coded with a value of 7, 23, or 27 depending if the sequence was part of a nested transaction, unnested, unchained transaction, or unnested, chained transaction, respectively. The different sub-addresses are used to alleviate any ambiguity that might occur. The retry sub-address always means to repeat the last sequence which the RT transferred on the bus. (See Figure 3.9.4.2-1).

3.9.4.3 Shut-Down Sequence

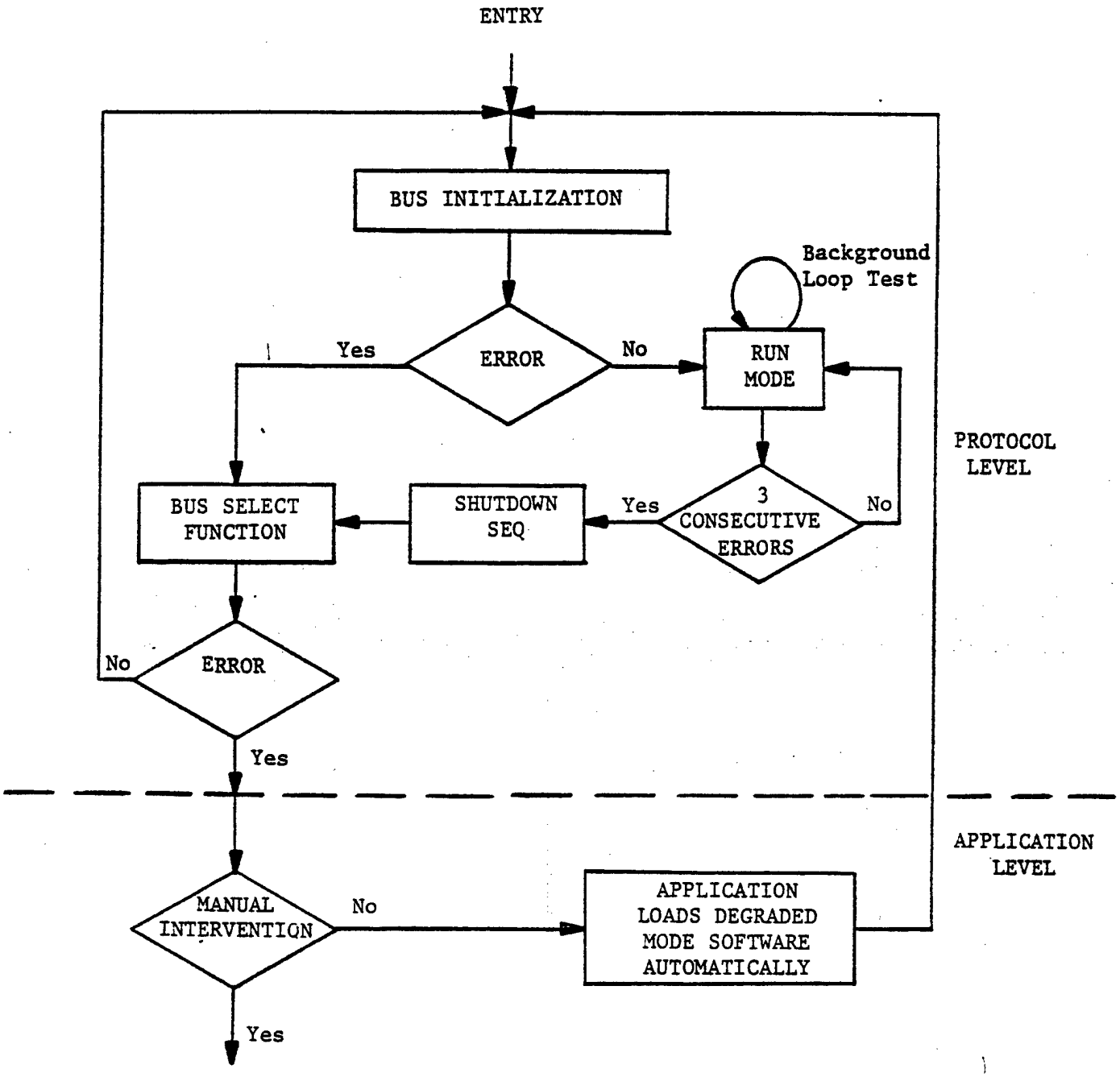
The following sequence will transfer control to the Bus Selection Function.

1. All transfers associated with the 2 RT's in question will be cancelled by the BC.

2. All new requests will be refused.

3. All existing active sequences, chassis and linkages will be permitted to run to completion.

4. At this point control will be transferred to the bus select function.



Manual Recovery Procedures. Available tools include loop test, continuous loop test and poll test.

FIGURE 3.9.4-1 ERROR RECOVERY TEST

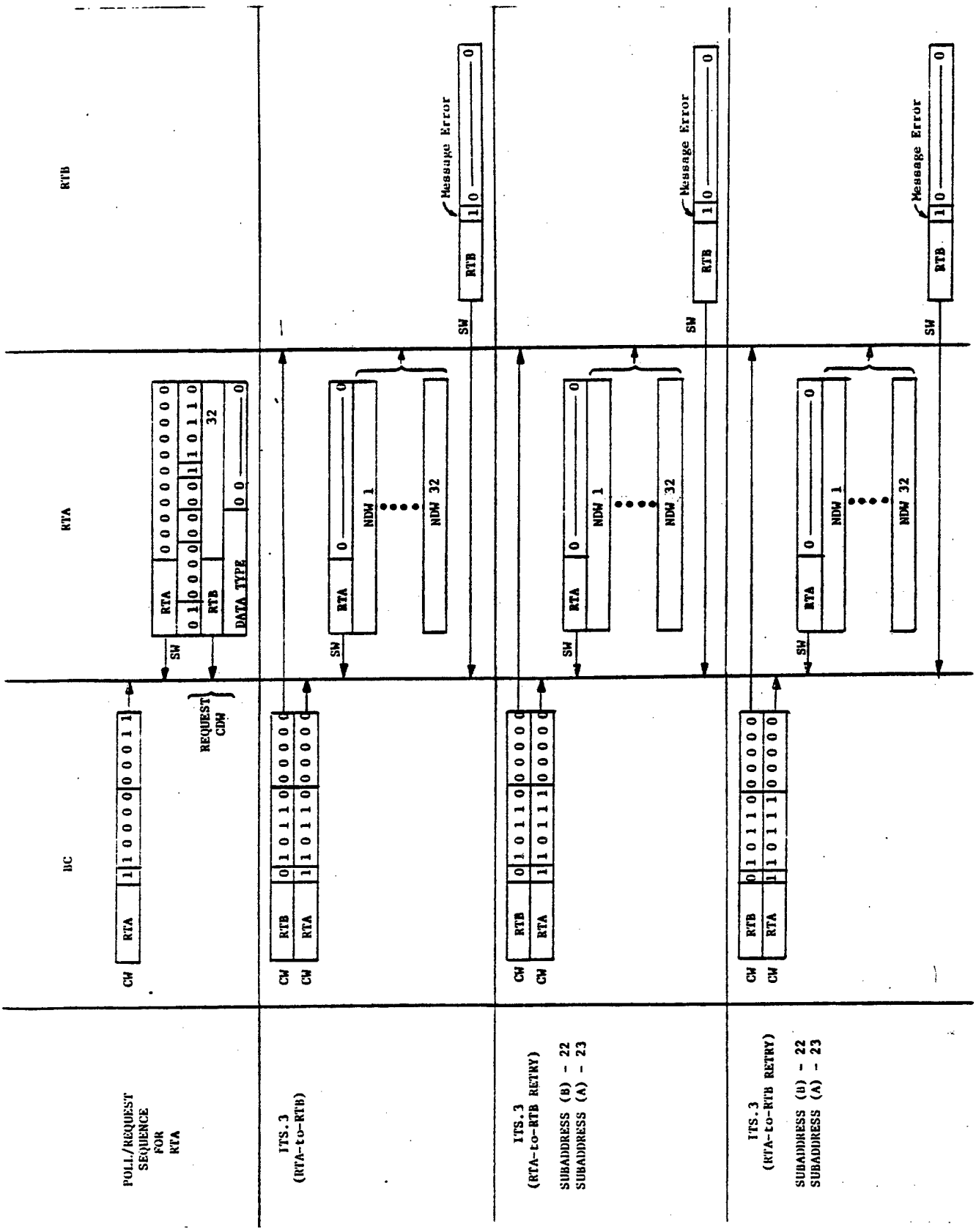


FIGURE 3.9.4.2-1 EXAMPLE OF ERROR RETRY